



אוניברסיטת בן-גוריון בנגב
הפקולטה למדעי ההנדסה
המחלקה להנדסת מכונות

אלגוריתם תכנון הליכה לרובוט ארבע רגלי

מאת : ים גבע

מנחה : ד"ר אמיר שפירא

_____ : תאריך

_____ : חתימת המחבר

_____ : תאריך

_____ : אישור המנחה

_____ : תאריך

_____ : אישור יו"ר ועדת תואר שני מחלקתית

תקציר

היכולת של כלי רובוטי לנייד עצמו במשטחים שונים הכרחית בפיתוח רובוטים הולכים כיום. מטרת המחקר היא פיתוח אלגוריתם גנרי אשר יהיה מסוגל לתכנן את שלבי ההליכה של כל רובוט הולך ארבע רגלי נתון על פני שטח מורכבים, כלומר, בהינתן מבנה הרובוט ונקודת יעד או מסלול ניווט רצוי, האלגוריתם יתכנן את סט הצעדים הדרוש על מנת לבצע את משימת הניווט הרצויה תוך שמירה על אילוצי קינמטיקת הרובוט, אילוצי שיווי המשקל ומאפייני תנועה רצויים הניתנים לקביעה ע"י המפעיל.

האלגוריתם מבוסס על שילוב של שני כלים עיקריים, האחד הוא שיטת חיפוש בגרף והשני הוא פונקציית פוטנציאל. שילוב זה מאפשר את תכנון משימת הניווט המוטלת על הרובוט תוך התמודדות עם החסרונות של כל כלי בנפרד. עבור כל צעד, האלגוריתם שוקל את כל אפשרויות ההתקדמות ובכך בונה את גרף החיפוש. פונקציית הפוטנציאל משמשת לצורך הערכת הצעדים על-מנת לבחור את הצעד המיטבי.

בשונה מאלגוריתמים דומים לתכנון הליכת רובוטים ארבע רגליים, האלגוריתם המוצע משתמש בקריטריון חדשני לצורך הבטחת שמירה על אילוצי שיווי המשקל של הרובוט ובכך מבטיח תכנון מסלול אמין עבור תנאי שטח מורכבים. קריטריון זה לוקח בחשבון את מקדם החיכוך בין רגל הרובוט לקרקע וכן את שיפועי הקרקע בנקודות המגע (ללא הנחת קרקע מישורית).

הסימולציות והניסויים שבוצעו במהלך המחקר מוכיחים את היתכנות ויתרון השימוש באלגוריתם לצורך תכנון תנועה בתנאי שטח מורכבים. הסימולציות אשר ממשות את האלגוריתם נכתבו בתוכנת MATLAB ותוצאותיהן מכילות המחשה גרפית של פלטי האלגוריתם. הניסויים שבוצעו כחלק ממחקר זה בוצעו תוך שימוש ברובוט הולך ארבע אשר תוכנן ונבנה כחלק מהמחקר. תוצאות ומסקנות המחקר פורסמו במאמר והוצגו במספר כנסים.

בעבודה זו מוצגת סקירה מקצועית של הרקע התאורטי אשר עליו מבוססת העבודה. כמו כן, מוצג האלגוריתם המפותח תחת מחקר זה, תוצאות סימולציות וניסויים הממחישים ומתארים את אופן פעולתו וכן תיאור הרובוט שתוכנן ונבנה לצרכי המחקר.

תוכן עניינים

1	מבוא	1
2	סקר ספרות	2
2.1	רובוטיים הולכי ארבע	2
2.2	תכנון הליכת רובוטים רגליים	3
2.3	הגדרת יציבות ברובוטים הולכים	5
2.3.1	הגדרות מקדימות	5
2.3.2	הגדרות קריטריוני יציבות	5
2.4	ניווט רובוטים	7
2.4.1	פונקציית פוטנציאל	7
2.4.2	חיפוש בגרף	8
3	רובוט הולך ארבע רגלי – תכן, מידול וקינמטיקה	11
3.1	תכן הרובוט	12
3.2	בקרה ותקשורת	15
3.3	מידול הרובוט	17
3.4	קינמטיקה ישירה - Forward Kinematics	18
3.5	קינמטיקה הפוכה - Inverse Kinematics	20
3.6	מרחב העבודה של גוף הרובוט – <i>CKW (COG Kinematics Workspace)</i>	22
3.7	מרחב העבודה של רגלי הרובוט – <i>LWV (Leg Workspace Volume)</i>	23
4	מידול הקרקע, ויציבות הרובוט	25
4.1	מודל הקרקע	25
4.2	קריטריון היציבות - <i>FER (Feasible Equilibrium Region)</i>	26
4.3	יציבות הרובוט	29
5	אלגוריתם לתכנון הליכה	32
6	סימולציות	39
6.1	מבנה עקרוני של התוכנה המממשת את האלגוריתם	39
6.2	תוצאות סימולציות	40
6.2.1	סימולציה I	41

42.....	סימולציה II	6.2.2
45.....	סימולציה III	6.2.3
58.....	סימולציה IV	6.2.4
60.....	סימולציה V	6.2.5
63.....	7 ניסויים	
63.....	מערכת המדידה החיצונית	7.1
65.....	תוצאות ניסויים	7.2
65.....	ניסויי וידוא קריטריון יציבות <i>FER</i>	7.2.1
67.....	ניסוי הליכה על קרקע מישורית	7.2.2
69.....	ניסוי הליכה על קרקע מורכבת	7.2.3
72.....	8 סיכום ומסקנות	
73.....	רשימת מקורות	
78.....	נספח A – פירוט על פונקציות MATLAB עיקריות	
82.....	נספח B – שרטוטי חלקים שתוכננו לשם בניית הרובוט	
86.....	נספח C – מפרט מנועי RX-64	
87.....	נספח D – מפרט בקר ARBOTIX	
88.....	נספח E – מפרט רכיבי התקשורת האלחוטית	

רשימת איורים

איור 1 - רובוטים הולכי ארבע.....	3
איור 2 - הגדרת קריטריוני היציבות SSM, LSM, CLSM. נלקח מ- [28].....	6
איור 3 - קריטריון היציבות NESM.....	6
איור 4 - (a) המחשה גרפית של פוטנציאל משיכה, (b) גרדיאנט של פוטנציאל משיכה.....	7
איור 5 - בעיית המינימום המקומי.....	8
איור 6 - מימין: קשתות ללא כיווניות, משמאל: קשתות בעלות כיוון.....	9
איור 7 - אופן חיפוש אלגוריתם ה-DFS.....	10
איור 8 - תמונה של הרובוט הבנוי.....	11
איור 9 - מבנה רגלי הרובוט.....	12
איור 10 - המחשה למקרה קיצון בעומסי המפרקים.....	12
איור 11 - מבנה הרובוט. (a) מבט צד, (b) מבט על.....	13
איור 12 - תרשים סכמתי של התקשורת והאלקטרוניקה ברובוט.....	16
איור 13 - מודל הרובוט.....	17
איור 14 - מרחב העבודה של גוף הרובוט - CKW.....	22
איור 15 - מרחב העבודה של רגל הרובוט LWV – (a) מבט צד, (b) מבט תלת ממדי.....	23
איור 16 - מודל הקרקע.....	25
איור 17 - קירוב קונוס החיכוך לקונוס פוליהדרלי חסום.....	27
איור 18 - שטח ה- FER עבור ערכי N שונים.....	27
איור 19 - דוגמא לשטח FER היוצא מגבולת ה- SP	28
איור 20 - היטל אנכי של פרמטרי הרובוט עבור $ML=4$	29
איור 21 - שמונת הקומבינציות האפשריות לצעד כפול.....	33
איור 22 - תרשים זרימה מפורט של האלגוריתם.....	36
איור 23 - ממשק משתמש לקביעת הגדרות.....	40
איור 24 - תוצאות סימולציה I – פלט האלגוריתם.....	41
איור 25 - תוצאות סימולציה I - שינוי הפוטנציאל.....	42
איור 26 - תוצאות סימולציה II – פלט האלגוריתם לפני הרצת השלב השני.....	43
איור 27 - תוצאות סימולציה II – שינוי הפוטנציאל לפני הרצת השלב השני.....	43
איור 28 - תוצאות סימולציה II – פלט האלגוריתם אחרי הרצת השלב השני.....	44
איור 29 - סימולציה III – מודל הקרקע.....	45
איור 30 - תוצאות סימולציה III – מסלול מרכז המסה וה- BP עבור הרצה עם קריטריוני יציבות שונים.....	46
איור 31 - תוצאות סימולציה III – הצגת התוצאות, הקרקע, נקודת המוצא ונקודת היעד.....	46
איור 32 - תוצאות סימולציה III – שינוי הפוטנציאל עבור הרצה עם קריטריוני יציבות שונים.....	47
איור 33 - תנועות הרובוט בהרצה עם קריטריון SSM (מבט על).....	49

57.....	איור 34 - תנועות הרובוט בהרצה עם קריטריון <i>FER</i> (מבט על)
58.....	איור 35 - תוצאות סימולציה IV – מסלול מרכז המסה – מבט על
59.....	איור 36 - תוצאות סימולציה IV – מסלול מרכז המסה – מבט צד
59.....	איור 37 - תוצאות סימולציה IV – מסלול מרכז המסה – מבט תלת-ממדי
60.....	איור 38 - תוצאות סימולציה IV – שינוי הפוטנציאל לאורך התקדמות הרובוט
61.....	איור 39 - השפעת צפיפות מרחב ה-CKW על זמן הריצה כאשר אילוץ C מבוטל
61.....	איור 40 - השפעת צפיפות מרחב ה-CKW על זמן הריצה כאשר אילוץ C מופעל
62.....	איור 41 - השפעת רזולוציית הגריד על זמן הריצה
63.....	איור 42 - אילוסטרציה למיקום מצלמות סביב זירת ניסוי
64.....	איור 43 - הארכיטקטורה המשמשת למעבר המידע ברובוט
65.....	איור 44 – מיקום וזוויות גוף הרובוט כתלות בזמן
66.....	איור 45 - מיקום ההיטל הופקי של מרכז המסה ברגע הנפילה
66.....	איור 46 - מיקומי היטלי מרכז המסה ברגע הנפילה ב-24 ניסויים
67.....	איור 47 - מיקומי היטלי מרכז המסה ברגע הנפילה ב-24 ניסויים - הגדלת אזור הנפילות
68.....	איור 48 - תוצאות ניסויי הליכה בחוג פתוח וחוג סגור
69.....	איור 49 - ניסוי הליכה על קרקע מורכבת - קרקע הניסוי
	איור 50 - תוצאות ניסוי הליכה על קרקע מורכבת עם קריטריון <i>SSM</i> בלבד – מיקום
69.....	ואוריינטציית גוף הרובוט כתלות בזמן
	איור 51 - תוצאות ניסוי הליכה על קרקע מורכבת עם קריטריון <i>SSM</i> בלבד – מיקום מצוי אל
70.....	מול רצוי
	איור 52 - תוצאות ניסוי הליכה על קרקע מורכבת עם קריטריון <i>FER</i> - מיקום ואוריינטציית גוף
70.....	הרובוט כתלות בזמן
71.....	איור 53 - תוצאות ניסוי הליכה על קרקע מורכבת עם קריטריון <i>FER</i> - מיקום מצוי אל מול רצוי

רשימת טבלאות

14.....	טבלה 1 - ממדי ופרמטרי התכן של הרובוט
39.....	טבלה 2 - הפרמטרים העיקריים המשפיעים על אופן ריצת האלגוריתם
78.....	טבלה 3 - פירוט על פונקציות MATLAB עיקריות

רשימת הסימנים

<u>Symbol</u>	<u>Description</u>
x_i	Contact point i position vector
f_i	Contact point i reaction force vector
f_g	Gravity vector
x_c	Center of gravity position vector
n_i	Normal vector at contact point i
μ_i	Coefficient of friction at contact point i
x_0, y_0, z_0	World attached axes
x_c, y_c, z_c	Robot body attached axes
x_B, y_B, z_B	Scapula joints attached axes
x_i, y_i, z_i	Link i Reference frame axes
θ_i	Actuator I angle
q	Robot configuration vector
L	Footholds position vector
L_i	Leg i position vector
COG	Center Of Gravity configuration vector
φ_{roll}	Robot body roll angle in relation to x_0
φ_{pitch}	Robot body pitch angle in relation to y_0
φ_{yaw}	Robot body yaw angle in relation to z_0
B	Scapula joints position vector
B_i	Leg i scapula joint position vector
MS	Minimum SSM
FSF	FER safety factor
ML	Moving Leg index
NML	Next Moving Leg index
MSA_i	Margin of Stability Area for $ML=i$
CKW	COG Kinematics Workspace
CSW_i	COG Stable Workspace for $ML=i$
LWV	Leg Workspace Volume
LWA	Leg Workspace Area
BP	Body Position

U	Potential function
FER	Feasible Equilibrium Region
SP	Support polygon
LPT	Low Potential Threshold

1 מבוא

רובוטים רגליים מתאפיינים בעבירות וביכולת תמרון גבוהה ביחס לרכבים גלגליים. אם כי בקרת רכבים גלגליים פשוטה יותר בהשוואה לרובוטים רגליים, ניתן להפעיל רכבים גלגליים בעיקר בדרכים סלולות ולא בתנאי שטח קשים – ביערות, חולות ובאזורי אסון. במקרים אלו, השימוש ברובוטים עם גלגלים מצטמצם עד מאוד והופך לבלתי אפשרי ולכן מועדפים רובוטים רגליים. מסלולי ההליכה של רובוטים אלו בדידים, כלומר רגלי הרובוט מונחות על נקודות מסוימות על המסלול ואין דרישת רציפות המסלול שקיימת עבור רכבים גלגליים. דוגמאות למשימות בהן קיימים יתרונות ניתן למצוא בתחום החילוץ וההצלה וכן המשימות נשיאת משאות.

עיקר הבעיה בשימוש ברובוטים רגליים היא שמירה על היציבות/שיווי משקל בזמן התנועה. נושא זה הוא מורכב משום שיש לקחת בחשבון את מבנה הרובוט, צורת ההליכה, סוג המסלול ותוואי שטח. המטרה היא פיתוח אלגוריתם אשר יהיה מסוגל לתכנן את סט הצעדים הדרוש על מנת שהרובוט יוכל לבצע את משימת הניווט הרצויה. הדגש בתכנון האלגוריתם הוא על יכולת התמודדות עם תנאי שטח קשים ומורכבים במיוחד. הכוונה בתנאי שטח מורכבים היא לאי אחידות, שיפועים גדולים, בורות, ומקדמי חיכוך נמוכים. שיטת הפתרון שנבחרה היא שילוב של שימוש בשיטת DFS (Depth First Search) לחיפוש בגרף ובפונקציית פוטנציאל (APF - Artificial Potential Field). האלגוריתם משתמש בכלים הנ"ל לצורך תכנון רצף צעדים לא מחזורי אשר משמש להנעת הרובוט לעבר נקודות ציון רצויות. באמצעות ניתוח תוואי הקרקע, האילוצים הקינמטיים של הרובוט וקריטריון שיווי משקל האלגוריתם מייצר את רצף הצעדים אשר יבטיח תנועה יציבה של הרובוט.

התרומה העיקרית של האלגוריתם המוצג לעומת אלגוריתמים דומים לתכנון תנועה של רובוטים הולכי ארבע היא האופן בו האלגוריתם מבטיח שמירה על שיווי משקל. לצורך הערכה כמותית של שיווי המשקל האלגוריתם מבצע שימוש בקריטריון יציבות חדשני. קריטריון זה אינו מבוסס על הנחות כלשהן עבור תוואי הקרקע (כגון הנחת קרקע מישורית) ולכן מאפשר תכנון תנועה גם על קרקע מורכבת במיוחד תוך התחשבות במאפיינים הגיאומטריים של פני השטח ובמקדם החיכוך בין כפות רגלי הרובוט לקרקע. תרומה נוספת של מחקר זה היא פיתוח רובוט הולך ארבע רגלי. הרובוט תוכנן ויוצר לצורך ביצוע ניסויים לאימות האלגוריתם וכן לצרכי מחקר עתידיים.

בעבודה זו אציג רקע תאורטי בתחום ההליכה של רובוטים ארבע רגליים, אפרט על אופן מידול הבעיה, אציג את האלגוריתם לפתרון הבעיה וכן אציג תוצאות סימולציות וניסויים.

2 סקר ספרות

שלב הנחת היסודות התיאורטיים הינו שלב חשוב המאפשר רקע איתן דיו להבנת הבעיה המוצגת במחקר. הסקירה הספרותית מאפשרת הבנה של הרציונל העומד מאחורי הבעיה: מוטיבציה למציאת פתרון לבעיה, התפתחות הבעיה והתקדמות בפתרונה לאורך השנים.

בתחילת פרק זה מוצגת סקירה של רובוטים הולכי ארבע בעלי מאפיינים דומים לאלו הנדרשים מהרובוט שפותח כחלק ממחקר זה. לאחר מכן מוצגות ההגדרות הבסיסיות בנושא הליכת רובוטים רגליים. בהמשך מוצגים אופנים וקריטריונים להגדרת יציבות של רובוטים הולכים. לבסוף, מוצגת סקירה בתחום ניווט רובוטים הרלוונטי למחקר זה.

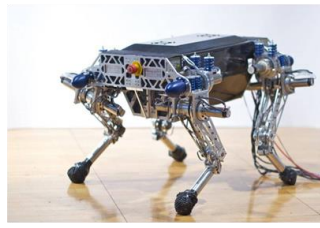
2.1 רובוטיים הולכי ארבע

חקירת הליכה ומחקרים הקשורים למכונות הולכות החלו במאה ה-19, כאשר מחקר של אחד הדגמים הראשונים פורסם ב-1870 [1]. במשך 80-90 השנים הבאות הושקעו מאמצים לבנות מכונות הולכות המבוססות על שרשראות קינמטיות שונות שהיו אמורות לגרום לתנועות רצויות תוך כדי הפעלתם. בשנים אלו הוצעו דגמים רבים, אך הביצועים של המכונות האלו היו מוגבלים ע"י צורות הליכה קבועות, ואי יכולת להתאים את ההליכה לשינוי בתוואי השטח. בסוף שנות ה-50 של המאה ה-20 התברר כי לא ניתן לתכנן מכונות הולכות המבוססות על מנגנונים קינמטיים בלבד המספקים תנועות קבועות בלבד וכי יש צורך לשלב מנגנוני בקרה ותכנון. בשנת 1966 נבנה ע"י R.B.McGhee ו-A.A. Frank [2] הרובוט הראשון שנע בצורה אוטונומית בעזרת בקרה ממוחשבת והנעה חשמלית. המשימה העיקרית של המחשב של הרובוט הייתה לפתור את המשוואות הקינמטיות על מנת להפעיל את המנועים החשמליים שהניעו את הרגליים כך שהרובוט יתקדם תוך שמירה על אילוצי שיווי משקל. מאז ועד היום, בעקבות התקדמות הטכנולוגיה בנושאי הבקרה, המחשוב וההנעה, נבנו רובוטים הולכים רבים בעלי יכולות שונות ומגוונות כגון ריצה [3], הליכה בתנאי שטח קשים, קפיצה מעל מכשולים, טיפוס [4] [5] ועוד.

כיום, קבוצות מחקר רבות מתכננות ובונות רובוטים משלהן לצורך אימות האלגוריתמים והתאוריות אותם הם מפתחות. דוגמאות לרובוטים כאלו המפותחים לעבודה בתוך מעבדה הם ה-LittleDog של Boston Dynamics [6], ה-StarlETH [7] ו-ALoF [8] של ETH, TITAN-IX [9], MRWALLSPECT III [10], AiDIN-III של אוניברסיטת Sungkyunkwan (SKKU), רובוט דמוי עכביש [11] של הטכניון ועוד. איור 1 מציג דגמים אלו.



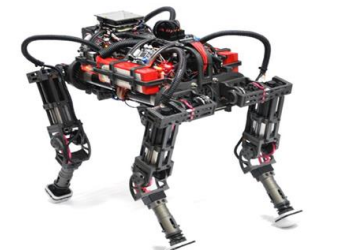
AloF



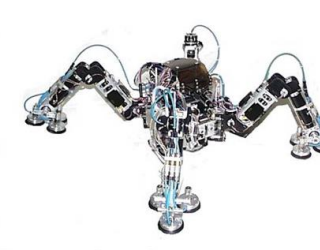
StarLETH



LittleDog



AiDIN-III



MRWALLSPECT-III



TITAN-IX

איור 1 - רובוטים הולכי ארבע

לרובוטים אלו יחידות עיבוד חזקות, מבנה מורכב או חיישנים המורכבים על הרובוט. מכיוון שבדרך כלל מטרת המחקר היא לא פיתוח הרובוט עצמו והרובוט משמש ככלי עזר למחקר, ברוב המקרים הרובוט המפותח הינו בעל חומרה ותוכנה מורכבים. עובדה זו מקשה על ייצור הרובוט על-ידי קבוצת מחקר אחת.

הרובוט אשר תוכנן כחלק מעבודת המחקר הינו בעל מאפיינים אשר מאפשרים ייצור ותפעול פשוט ומהיר. בניגוד לרובוטים בעלי חומרה ותוכנה מורכבים, הרובוט בנוי במחלקי מדף וניתן לבנייה בעלות נמוכה יחסית. כמו כן, התכן הוא מודולארי ומאפשר שינוי מבנה גם לאחר הבנייה הראשונית. כל התכונות הנ"ל מביאות לרובוט אשר יכול למש כפלטפורמה אוניברסלית למחקר הליכת רובוטים הולכי ארבע. פירוט נרחב על הרובוט שתוכנן כחלק ממחקר זה נמצא בפרק 3.

2.2 תכנון הליכת רובוטים רגליים

כיום קיימות מספר רב של קבוצות אשר חוקרות את תחום הליכת רובוטים רגליים [12], [13], [14], [15], [16], [17], [11], [18], [19]. מלבד תכנון הרובוט עצמו, האתגר העיקרי הוא תכנון הליכת הרובוט. כלומר, תכנון צעדי הרובוט כך שתתקבל תנועה רצויה תוך שמירה על אילוצי שיווי המשקל או היציבות. קיימות שתי גישות עיקריות לתכנון תנועה של רובוטים הולכים. הגישה הראשונה, "הליכה דינאמית", בה במהלך שלבי ההליכה לרובוט יציבות מחזורית ותכנון ההליכה מתחשב בכוחות האינרציה והכוחות החיצוניים. גישה זו מאופיינת בתנועה רציפה ומהירה וכן במגבלות וקושי להתמודדות עם תנאי קרקע מורכבים עקב מחזור הליכה קבוע בעל יכולת הסתגלות נמוכה לשינויים. הגישה השנייה, "הליכה קוואסי-סטטית", בה במהלך כל שלבי ההליכה הרובוט שומר על שיווי משקל סטטי. גישה זו מאופיינת בהליכה איטית יותר אך בעלת יכולת עבירות גבוהה יותר בתנאי קרקע מורכבים – האלגוריתם בפרויקט זה מתבסס על שיטה זו לתכנון ההליכה.

סיווג נוסף לתכנון הליכה הינו אופן קביעת סדר תנועת הרגליים והחלטה על מיקום הדריכה על הקרקע במהלך ההליכה. שיטה אחת היא השיטה ה"מחזורית" (Periodic gait pattern) המאופיינת בסדר קבוע של תנועת רגליים ובפרופיל תנועה קבוע להזזת הרגליים. קיימות תבניות קבועות לתנועה מסוג זה אשר מתאימות לאופני תנועה רצויים כגון הליכה או פנייה. היתרונות בשימוש בשיטה זו הוא בכך שאין צורך בחישובים מורכבים וכן היכולת לנתח את התבניות הקבועות ולבצע אופטימיזציה על פרמטרי ההליכה.

החיסרון העיקרי של השיטה נובע מהפשטות שלה, לצורת הליכה זו יכולות מוגבלות מאוד בהליכה על תוואי שטח מורכב ולא קיימת יכולת הסתגלות לשינויים בתוואי השטח. השיטה השנייה, השיטה ה"חופשית" (Free gait או Non-periodic gait) בה לא קיים תכנון מוקדם לאופן ההליכה ואופן ההליכה נקבע באמצעות סט של חוקים או באמצעות שיטת חיפוש. חוקים אלו יכולים להיקבע מראש ע"י המשתמש [20], [16], [21], [22], להילמד באופן אוטומטי [23] או להיקבע בהשפעת בעלי-חיים בטבע [24]. בשיטת החיפוש האלגוריתם האחראי לתכנון ההליכה מייצר אוסף של צורות תנועה אפשריות ובאמצעות השוואה המבוססת על פרמטרים הניתנים לשינוי, נבחרת התנועה המתאימה ביותר לצעד הנוכחי או סט התנועות הבאות במידה ומתבצע תכנון של יותר מצעד אחד קדימה. החסרונות המהותיים ביותר של השיטה הם הסיבוכיות הגבוהה והצורך ביכולות חישוב מתקדמות עקב הצורך בחיפוש התנועה האופטימלית מבין מבחר גדול מאוד של תנועות אפשריות [4]. חסרון נוסף בשיטה זו הוא הצורך במידע לגבי תוואי השטח עליו הולך הרובוט. למרות חסרונות אלו, שיטת החיפוש נבחרה לשימוש בפרויקט זה עקב היכולת שלה להתמודד עם תוואי שטח מורכב ומשתנה וכן עקב העובדה שיכולות החישוב מתפתחות בקצב גבוה וקיימים חיישנים המאפשרים לבצע מיפוי שטח באיכות גבוהה.

קיימת צורה נוספת לסיווג הליכת רובוטים המתייחסת לרציפות התנועה. ב"הליכה רציפה" (Continuous walking) [16] גוף הרובוט נע באופן רציף בזמן ההליכה, כלומר, גוף הרובוט נע במקביל לתנועת הרגליים. ב-"הליכה מקוטעת" (Discontinuous walking) [21], [25], תנועת גוף הרובוט ותנועת הרגליים מתבצעות בשני פרקי זמן שונים וניתן לחלק את התנועה לשלב תנועת מרכז המסה ולשלב תנועת הרגל. האלגוריתם בפרויקט זה משתמש בשילוב של שתי השיטות הנ"ל [15]. שילוב זה מתבסס על שיטת תכנון של "צעד כפול", כלומר, גוף הרובוט מבצע תנועה בין כל שני צעדים עוקבים. שילוב זה מחייב את מיקום מרכז הכובד להיות במקום אשר יבטיח שיווי משקל לשני הצעדים העוקבים ולא רק לצעד הבא כמו ב"הליכה מקוטעת".

2.3 הגדרת יציבות ברובוטים הולכים

על-מנת לתכנן הליכה של רובוט הולך יש לקחת בחשבון את יציבות הרובוט בכל שלב ושלב של תנועת הרובוט. כדי לקחת בחשבון שיקולי יציבות אלו, יש תחילה להגדיר מתי הרובוט יציב. החלוקה העיקרית בין הגדרות היציבות השונות נובעת מההבדל בין שיטות תכנון ההליכה. כאשר מתבצעת "הליכה דינאמית" קיימים קריטריונים אשר לוקחים בחשבון את כוחות האינרציה והכוחות החיצוניים הפועלים על הרובוט. לעומת זאת בהליכה "קוואסי-סטטית" הקריטריונים המשפיעים על היציבות מבוססים על משוואות שיווי המשקל של הרובוט. מכיוון ששיטת ההליכה שנבחרה לאלגוריתם היא הליכה "קוואסי-סטטית", בחלק הבא אציג קריטריונים שונים המשמשים להגדרת יציבות בשיטת הליכה זו. תחילה נגדיר מספר מושגים אשר ישמשו לצורך תיאור קריטריוני היציבות:

2.3.1 הגדרות מקדימות

מרכז הכובד (Center Of Gravity) COG – נקודה זו, הנקראת גם מרכז המסה של מערכת גופים או חלקיקים, מוגדרת כפונקציה של כל הגופים או החלקיקים המרכיבים את המערכת באופן הבא:

$$\overrightarrow{COG} = \sum_i \frac{m_i \vec{r}_i}{m_i} \quad (1)$$

כאשר m_i הוא מסת הגוף ו- \vec{r}_i הוא מיקומו. היתרון בשימוש במרכז הכובד הוא היכולת להתייחס לכל המערכת כמסה נקודתית הנמצאת בנקודה זו.

פוליגון התמיכה (Support Polygon) SP – זהו פוליגון המוגדר באמצעות הנקודות שמיקומן הוא מיקום כפות הרגליים של הרובוט.

כעת ניתן להציג מספר קריטריוני יציבות המשמשים לתכנון הליכה "קוואסי-סטטית". הקריטריונים הבאים מתייחסים למושג ה"יציבות" כמצב המתאר את שיווי המשקל של המערכת.

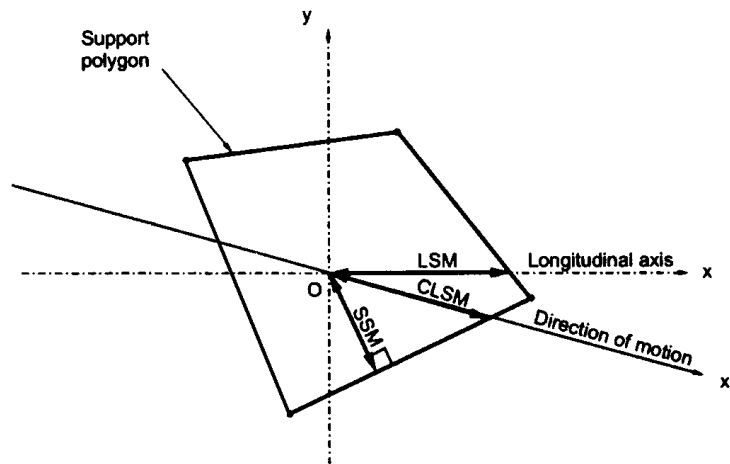
2.3.2 הגדרות קריטריוני יציבות

קריטריון יציבות SSM (Static Stability Margin) – לפי קריטריון זה, [2], המערכת בשיווי משקל אם המרחק של ההיטל האנכי של מרכז הכובד משפת פוליגון התמיכה (כאשר מרכז הכובד נמצא בתוך הפוליגון) גדול מאפס. מהגדרת קריטריון זה ניתן לקבל מדד לרמת היציבות של המערכת, מדד זה מוגדר כמרחק המינימלי של ההיטל האנכי של מרכז הכובד מצלעות הפוליגון (ראה איור 2). מרחק מינימלי זה מהווה מדד כמותי ביחידות אורך ליציבות המערכת. כאשר ערך מדד זה הינו אפס, כלומר, מרכז הכובד נמצא ממש מעל אחת מצלעות פוליגון התמיכה, המערכת נמצאת על סף יציבות.

קריטריון יציבות LSM (Longitudinal Stability Margin) – לפי קריטריון זה, [26], המערכת בשיווי משקל כאשר המרחק משפת הפוליגון של ההיטל האנכי של מרכז הכובד לאורך ציר ה-x גדול מאפס (ראה איור 2). קריטריון דומה לקריטריון הנ"ל, **CLSM (Crab Longitudinal Stability Margin)**, [27], מוגדר בצורה דומה רק שכעת הציר מוגדר באמצעות כיוון ההליכה של הרובוט (ראה איור 2). המדד ליציבות בקריטריונים אלו מוגדר באופן זהה כמרחק לשפת הפוליגון לאורך הציר אשר באמצעותו מוגדרת

היציבות. קריטריונים אלו פשוטים יותר לחישוב ונותנים הערכה טובה ל-SSM, אך אינם מתאימים לתוואי שטח שאינו מישורי.

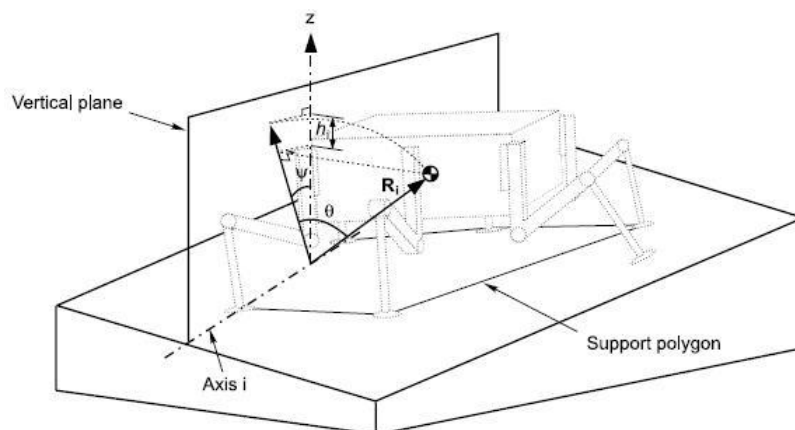
איור 2 מציג דוגמא לאופן חישוב מדדי היציבות הנ"ל בעת דריכה על ארבע רגליים כאשר LSM, CLSM ו-SSM הינם ממדי היציבות עבור כל אחד מהקריטריונים.



איור 2 - הגדרת קריטריוני היציבות SSM, LSM, CLSM. נלקח מ- [28]

קריטריון יציבות NESM (Normalized Energy Stability Margin) – בשונה מהקריטריונים

הקודמים, קריטריון זה, [28], לוקח בחשבון את גובה הרובוט ונותן מדד להפרש בין האנרגיה הפוטנציאלית הנוכחית לבין האנרגיה בזמן המעידה כאשר הצירים אשר סביבם מתבצעת המעידה הם צלעות פוליגון התמיכה. ככל שהפרש האנרגיות יותר גדול, כך הרובוט יציב יותר ולכן הממד מחושב לפי ההפרש המינימלי h_i כפי שמתואר באיור 3. זהו ערך ביחידות אורך הניתן לנרמול באמצעות הגובה המקסימלי אליו יכול להגיע הרובוט והוא פיתוח של הקריטריון המקורי, ESM [30], לאחר נרמול באמצעות משקל הרובוט. לקריטריון ESM מדד בעל יחידות אנרגיה ($mg h_i$) והוא מתאר את האנרגיה הפוטנציאלית המינימלית הדרושה "להפלת" הרובוט סביב צלעות פוליגון התמיכה. קיימות גרסאות נוספות של קריטריון זה הכוללות מודל מגע עם הקרקע וכן את השפעת הרגלים שאינן תומכות ברגע הערכת הקריטריון [31]. קריטריונים אלו המבוססים על שיקולי אנרגיה המתאימים לתנאי שטח אשר אינם מישוריים בלבד ומהווים פתרון איכותי לתכנון הליכה בתנאי שטח קשים.



איור 3 - קריטריון היציבות NESM

קריטריון יציבות (FER (Feasible Equilibrium Region) – כל הקריטריונים הנ"ל אינם לוקחים בחשבון תופעות אשר נובעות מכוחות החיכוך בנקודות המגע בין רגלי הרובוט לקרקע כמו החלקה. התרומה העיקרית של מחקר זה מבוססת על אינטגרציה של קריטריון יציבות אנליטי, [32], אשר מחשב את השטח המדויק אשר מעליו ניתן למקם את מרכז המסה. חישוב זה לוקח בחשבון את מקדם החיכוך בין רגל הרובוט לקרקע וכן את שיפועי הקרקע בנקודות המגע (ללא הנחת קרקע מישורית). השימוש בקריטריון זה מאפשר תכנון הליכה בתנאי שטח מורכבים במיוחד. הרחבה על אופן חישוב קריטריון זה מופיעה בפרק 4.2.

2.4 ניווט רובוטים

בפרק זה אציג רקע תאורטי על הכלים אשר באמצעותם מתכנן האלגוריתם את ניווט הרובוט.

2.4.1 פונקציית פוטנציאל

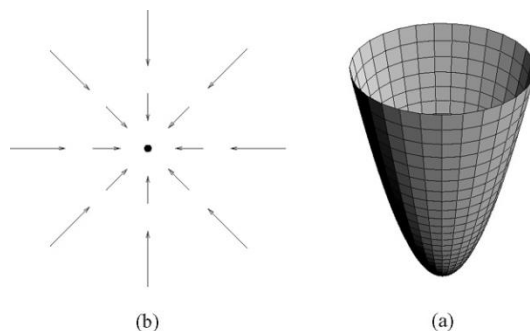
על מנת לבצע את משימת הניווט הרצויה, האלגוריתם משתמש בפונקציית פוטנציאל. פונקציית הפוטנציאל היא פונקציה גזירה המוגדרת בכל מרחב התנועה של הרובוט. נסמן את ערך הפונקציה כך:

$$U(q): \mathbb{R}^m \rightarrow \mathbb{R} \quad (2)$$

כאשר q מתאר את קונפיגורציית הרובוט ה- m ממדית. ערך הפונקציה שקול לאנרגיה, מכאן הגרדיאנט של פונקציית הפוטנציאל הינו וקטור הכוח המצביע לכיוון הפוטנציאל המקסימלי המקומי,

$$\nabla U(q) = DU(q)^T = \left[\frac{\partial U}{\partial q_1}(q), \dots, \frac{\partial U}{\partial q_m}(q) \right]^T \quad (3)$$

פונקציית הפוטנציאל מוגדרת כך שהכוח יכוון את הרובוט עד להגעה לנקודת היעד. בדרך כלל מגדירים את נקודת היעד כמקום עם הפוטנציאל הנמוך ביותר וכך ה"כוחות" הפועלים על הגוף בכל נקודה במרחב ימשכו אותו אל היעד כמו שניתן לראות באיור 4.



איור 4 - (a) המחשה גרפית של פוטנציאל משיכה, (b) גרדיאנט של פוטנציאל משיכה

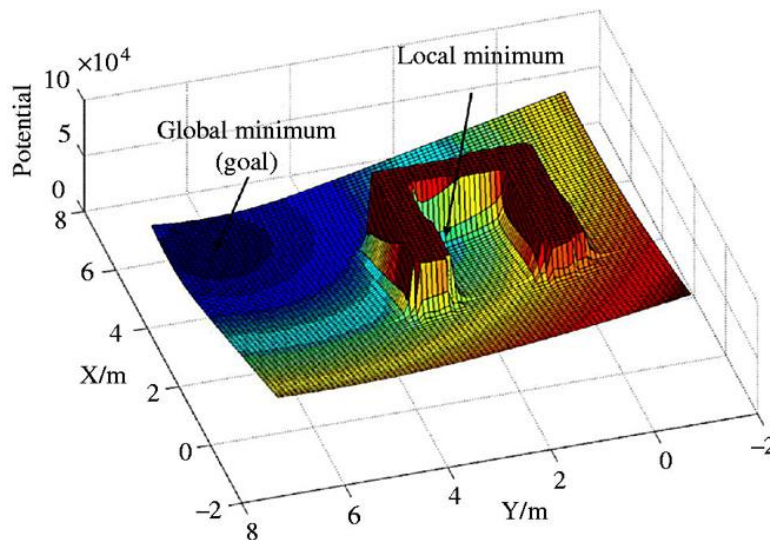
במרחב הכולל מכשולים ניתן להגדיר את המכשולים כאזורים בעלי פוטנציאל גבוה וכך ליצר כוחות דחייה אשר ירחיקו את הגוף המנווט מהמכשולים. כעת הפונקציה מכילה איבר משיכה ואיבר דחייה:

$$U(q) = U_{att}(q) + U_{rep}(q) \quad (4)$$

כאשר קיימות שיטות שונות להגדרת כוחות המשיכה, $U_{att}(q)$, וכוחות הדחייה, $U_{rep}(q)$.

השימוש בפונקציית פוטנציאל לתכנון הליכה בוצע בעבר [14], [13], לצורך הגדרה והתחמקות ממכשולים. באלגוריתם המפותח בפרויקט זה השימוש בפונקציית הפוטנציאל מתבצע לצורך בחירת הצעד אשר יקדם את הרובוט בצורה הטובה ביותר כאשר "צורה" זו מוגדרת באמצעות פונקציית הפוטנציאל כפי שיתואר בהרחבה בפרק 5.

אחת הבעיות הידועות בשימוש בפונקציית פוטנציאל היא בעיה נקודות המינימום המקומיות (Local Minima Problem [33]). כאשר קיימת נקודת מינימום נוספת המרחב אשר אינה נקודת היעד, יתכן מצב שבו הגוף המנווט יגיע לנקודה זו, שאינה הנקודה הסופית, ויישאר שם מכיוון שכל ערכי הפוטנציאל בסביבה הקרובה גבוהים יותר מהערך של המיקום הנוכחי (ראה דוגמא באיור 5).

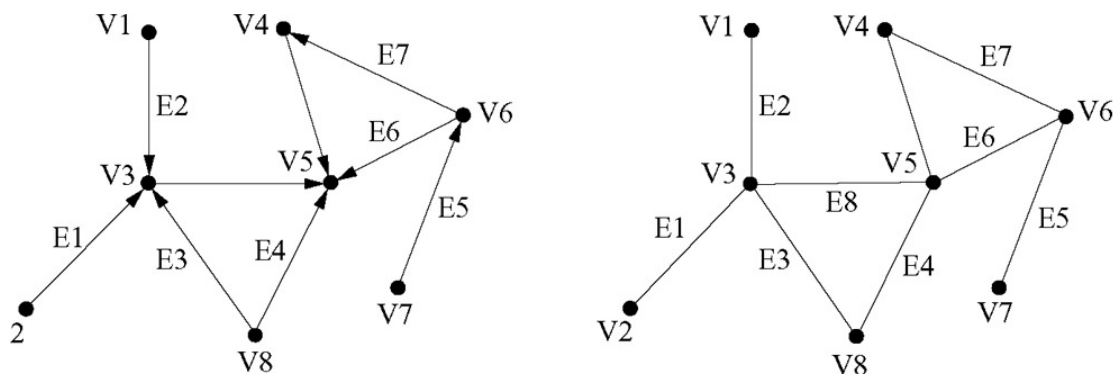


איור 5 - בעיית המינימום המקומי

השימוש בשיטת חיפוש בגרף מאפשר לאלגוריתם להתמודד עם בעיית המינימום המקומי. בנוסף, לאלגוריתם קיים שלב עיבוד נוסף אשר תפקידו לשפר את ביצועי האלגוריתם שנפגעו עקב תופעת המינימום המקומי. בפרק 5 המתאר את אופן פעולת האלגוריתם נושא ההתמודדות עם בעיה זו יוסבר בהרחבה.

2.4.2 חיפוש בגרף

השימוש בגרף לתכנון הליכה הוא כלי שנלמד בעבר [34]. הגרף הוא ייצוג מופשט של קבוצת אובייקטים והקישורים ביניהם. האובייקטים בגרף מיוצגים באמצעות צמתים (Nodes\Vertices) והקישורים מיוצגים באמצעות קשתות (Edges). הגרף משמש ככלי לניתוח בעיות מתמטיות בדידות. קשתות הגרף יכולות להיות בעלות כיוון או ללא כיווניות כפי שמתואר באיור 6. כאשר V מסמן צומת ו- E מסמן קשת.

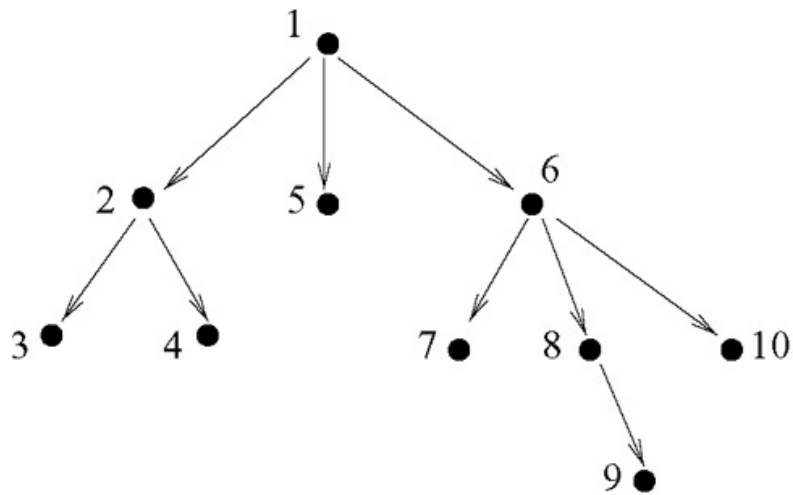


איור 6 - מימין: קשתות ללא כיווניות, משמאל: קשתות בעלות כיוון

קיימות שיטות רבות לביצוע חיפוש בגרף כאשר המטרה היא לחפש את המסלול (סט הקשתות) אשר בו יש לעבור מצומת לצומת כדי להגיע מהצומת ההתחלתית לצומת היעד. לדוגמא, ניתן להסתכל על הגרף כעל מפת דרכים אשר כל צומת בגרף מתאר צומת בדרכים וכל קשת היא שביל המחבר בין צמתים. ניתן לתת משקל לכל קשת על מנת אשר בדוגמא זו יהווה את אורך השביל. כעת האלגוריתם המבצע את החיפוש בגרף צריך לבחור את המסלול מצומת המקור לצומת היעד בעל האורך הכולל הקצר ביותר בדומה לפעולת מכשירי הניווט מבוססי ה-GPS המשתמשים לתכנון מסלול הנסיעה הקצר ביותר.

כל שיטת חיפוש מבוססת על אלגוריתם אשר בהינתן גרף סורק את הצמתים והקשתות על מנת לחפש את המסלול אל נקודת היעד. קיימים אלגוריתמים אופטימליים אשר סורקים את כל מרחב הגרף ומחפשים את המסלול האופטימלי. כאשר סריקה של כל המרחב אינה אפשרית, משום שלא קיים כל המידע או אם כי החישוב מצריך משאבי מחשוב רבים, ניתן להשתמש באלגוריתמים אשר אינם מבטיחים את קבלת המסלול האופטימלי אך כן מבטיחים הגעה לצומת היעד במידה וקיים מסלול.

אלגוריתם החיפוש הנבחר לשימוש בפרויקט זה הוא אלגוריתם ה-DFS (Depth-first search), [35]. במקור, אלגוריתם זה תוכנן לסרוק את כל מרחב הגרף אך מהשיקולים אותם ציינתי, ניתן להשתמש בו לחיפוש עד "עומק" מסוים אל תוך הגרף. בהרצת האלגוריתם בזמן אמת (הנעת הרובוט תוך כדי פעולת האלגוריתם), עומק זה מוגבל על ידי טווח הסריקה של חיישני סריקת השטח או על ידי המידע המקדים על תוואי השטח במידה וקיים כזה. איור 7 מתאר את אופן הפעולה של האלגוריתם.



איור 7 - אופן חיפוש אלגוריתם ה-DFS

המספרים באיור מתארים את סדר המעבר בין הצמתים. במקרה המתואר, הבחירה בצומת 2 כצומת הראשונה מתבצעת לפי החלטה מראש על כיוון הסריקה כאשר קיימת יותר מקשת אחת. ניתן להגדיר לכל קשת משקל וכך לבחור את הקשר הבאה להתקדמות לפי מדד מסוים. באלגוריתם במחקר זה, המשקל מחושב באמצעות פונקציית הפוטנציאל וכך הצומת בעלת הפוטנציאל הנמוך ביותר היא הצומת הבאה.

3 רובוט הולך ארבע רגלי – תכן, מידול וקינמטיקה

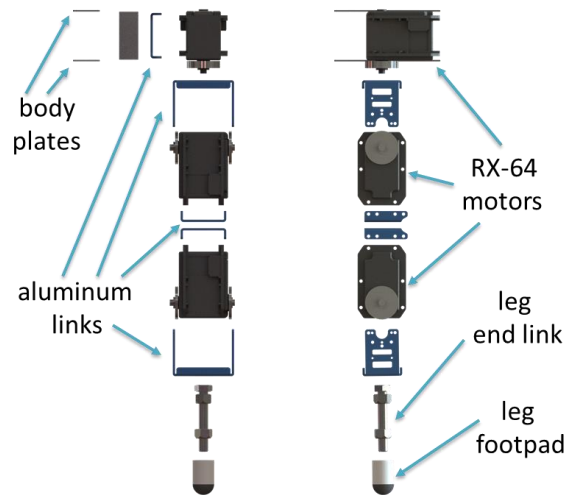
האלגוריתם שפותח במסגרת מחקר זה הינו גנרי, כלומר, הוא אינו מבוסס על קינמטיקת רובוט ספציפי וניתן להשתמש בו לצורך תכנון הליכה של כל רובוט בהינתן מבנה הרובוט. לצורך ביצוע ניסויים וכן לצרכי מחקר עתידיים, תוכנן ונבנה רובוט ארבע רגלי. בפרק זה מוצג תיאור התכן של הרובוט, אופן התקשורת עם הרובוט, מוצג פיתוח משוואות הקינמטיקה הישירה וההפוכה וכן ניתוח מרחבי העבודה של הרובוט. ההגדרות המתארות את מבנה הרובוט ומערכות הצירים ישמשו כבסיס לתיאור אופן פעולת האלגוריתם בפרק 5. פירוט מורחב יותר ניתן למצוא במאמר תכן שמכיל ניסויים נוספים המציגים את תפקוד הרובוט [35]. שרטוטים ומפרט רכיבים מופיעים בנספחים B-E.



איור 8 - תמונה של הרובוט הבנוי

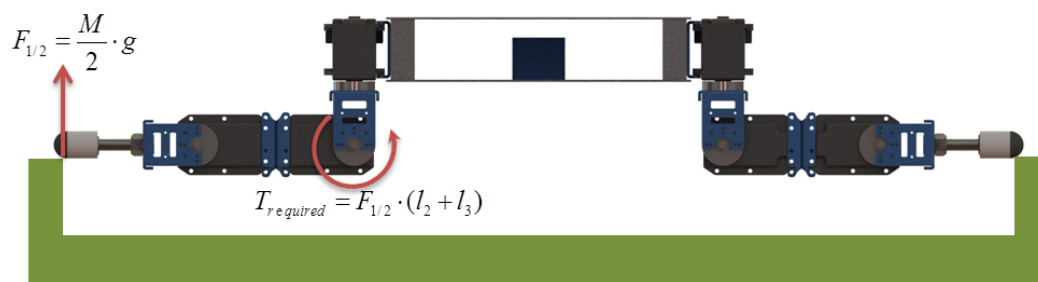
3.1 תכן הרובוט

הרובוט תוכנן באופן פשוט, מודולרי ומבוסס על חלקי מדף. כל אחת מארבעת רגלי הרובוט מורכבת משלושה מנועים חשמליים אשר מקנים 3 דרגות חופש לכל רגל. תכנון רגלי הרובוט מופיע באיור 9.



איור 9 - מבנה רגלי הרובוט

המנועים שנבחרו הינם מנועי סרוו מסוג Dynamixel RX-64 [35]. כל מנוע מסוגל לספק מומנט של [Kg- 64 cm]. טווח התנועה של כל מנוע הוא 300° והמהירות המקסימלית היא [RPM] 53.2. ניתן לקרוא בזמן אמת מכל מנוע את מהירותו, הזווית האבסולוטית בה הוא נמצא, טמפרטורה, מתח הפעלה ועומס. כל מנוע ניתן לתכנת בנפרד ולקבוע הגדרות כגון מהירות מקסימלית, עומס מקסימלי וכו'. מלבד חישובים קינמטיים, ניתן להשתמש בפיידבק מהמנועים הן לביצוע חישובי אנרגיה וכן ניתוח תופעות המגע עם הקרקע כמו למשל זיהוי רגע המגע של קצה הרגל עם הקרקע. השימוש במנועים אלו מאפשר לרובוט נשיאה של מטען נוסף במשקל של כ-2.5 ק"ג. חישוב משקל זה בוצע עבור המקרה האפשרי הגרוע ביותר בו הרובוט נתמך באמצעות שתי רגליים בלבד אשר פרוסות באופן אופקי. מקרה זה יתרחש כאשר הרובוט עומד על 3 רגליים כאשר כמעט כל המשקל נתמך רק באמצעות שתי רגליים. איור 10 מציג מקרה זה.



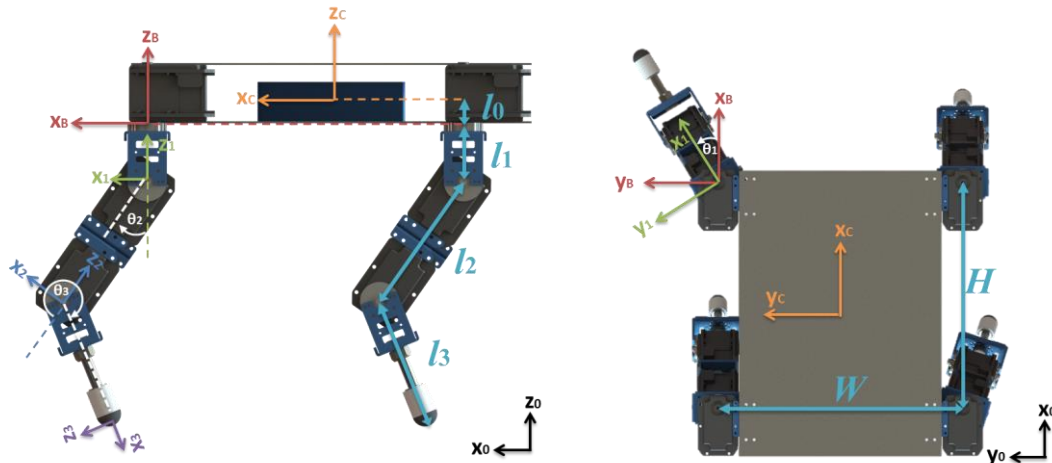
איור 10 - המחשה למקרה קיצון בעומסי המפרקים

איור 11 מציג את מבנה הרובוט, הסימונים עבור מידות הרובוט וכן את מערכות הצירים אשר ישמשו בהמשך לפיתוח משוואות הקינמטיקה של הרובוט. בשימוש הערכים והמידות מטבלה 1, מתקבל מומנט דרוש של $T_{required} = 3.5 [Nm]$. מנוע RX-64 יחיד מסוגל לספק $5.2 [Nm]$ בעוד שמנועי RX-28, דגם חלש

יותר מאותה סדרת מנועים, מסוגל לספק רק $3.67[Nm]$. על מנת לספק מרווח ביטחון נבחרו מנועי ה-RX-64. מנועים אלו נבחרו עבור כל 12 המפרקים משיקולי נוחות ומודולריות. התקשורת עם המנועים ובקרת המיקום מתבצעת באמצעות תקשורת סריאלית מסוג RS-485 ומנוהלת בבקר מובנה הנמצא בכל מנוע. הימצאות בקר בכל מנוע מביא למערכת מבוזרת אשר מאפשרת לבקר הראשי של הרובוט לבצע פעולות ברמה גבוהה יותר. המנועים מחוברים ביניהם ואל גוף הרובוט באמצעות חוליות אלומיניום סטנדרטיות [36]. חוליות אלו הן מוצר מדף. ניתן להשתמש בחוליות בעלי מבנה וגודל שונה כדי לשנות את מידות הרובוט וקונפיגורציית דרגות החופש בכל רגל.

חוליות הקצה בכל רגל הן בורג סטנדרטי מסוג M8x1.25x50 אשר מהודקת לחוליות האלומיניום באמצעות אומים. ניתן להשתמש גופים שונים ככפות רגליים, ברובוט שנבנה כפות הרגליים הן גלילי פלסטיק בעלי ראש גומי עגול אשר הודפסו במדפסת תלת-ממדית. ניתן להחליף את חוליות הקצה בחוליות מורכבות יותר המכילות קפיצים ו/או מרסנים לצורך הקניית מאפיינים רצויים לרגלי הרובוט.

גוף הרובוט מורכב משתי פלטות אלומיניום מקבילות בעובי $2 [mm]$. הפלטות מחוברות ביניהן באמצעות ארבע בלוקי אלומיניום בעלי חורים המתאימים לקיבוע חוליות האלומיניום באופנים שונים. תכן זה מאפשר חיבור מודולרי של דרגת החופש העליונה בכל רגל. הפלטה התחתונה מכילה חורים שנועדו לקיבוע הסוללה ורכיבי האלקטרוניקה בתוך גוף הרובוט, ראה איור 11.



איור 11 - מבנה הרובוט. (a) מבט צד, (b) מבט על

מידות ופרמטרי תכן נוספים של הרובוט שנבנה מופיעים בטבלה 1.

טבלה 1 - ממדי ופרמטרי התכן של הרובוט

תיאור	סימן	ערך
רוחב חזית גוף הרובוט	W	24.5 ס"מ
אורך צד גוף הרובוט	H	22.5 ס"מ
מרחק מרכז הכובד מפלטת הגוף התחתונה	l_0	2 ס"מ
אורך חוליית הרגל הראשונה	l_1	4 ס"מ
אורך חוליית הרגל השנייה	l_2	10.5 ס"מ
אורך חוליית הרגל השלישית	l_3	10.5 ס"מ
מגבלת תכן לדרגת החופש הראשונה	θ_{1_Limit}	$\pm 150^\circ$
מגבלת תכן לדרגת החופש השנייה	θ_{2_Limit}	$\pm 112^\circ$
מגבלת תכן לדרגת החופש השלישית	θ_{3_Limit}	$\pm 112^\circ$
משקל הרובוט (כולל סוללה)	M	3.4 ק"ג

מידות אלו מאפשרות לכפות רגלי הרובוט להגיע לשטח דסקה בקוטר 20 ס"מ על פני הקרקע בעוד שגוף הרובוט נמצא אופקי ובגובה 20 ס"מ מעל הקרקע. בנוסף, ניתן למקם את כף רגל הרובוט מעל מכשולים בגובה של 21 ס"מ. על מנת לעבור מכשול כזה יש לתכנן בקפידה את מסלול כל רגל וכן את תנועת גוף הרובוט על מנת לשמור על שיווי משקל.

שילוב הפרמטרים הנ"ל מאפשר הנחת רגליים חסרות משקל, כלומר, מיקום רגלי הרובוט משפיע באופן זניח על מיקום מרכז המסה של הרובוט כולו. הנחה זו מקלה באופן משמעותי על מורכבות חישובי אלגוריתמים לתכנון הליכה אשר צריכים להתייחס למיקום מרכז המסה של הרובוט בעת תכנון שלבי התנועה. השינוי במיקום מרכז המסה הנובע משינויים במיקום רגלי הרובוט אינו עולה על מידות ספירה ברדיוס 2.2 ס"מ. חישוב ערך זה נעשה באמצעות מידול הרובוט בתוכנת SolidWorks ומתן תכונות מסה מתאימות עבור כל חלק. חשוב לציין שאת הנחה זו יש לבחון מחדש לאחר כל שינוי תכן. בנוסף, כאשר מתבצעת תנועה מהירה יש לבחון את השפעת האינרציה של רגלי הרובוט על אפקטים דינמיים בתנועת הרובוט.

3.2 בקרה ותקשורת

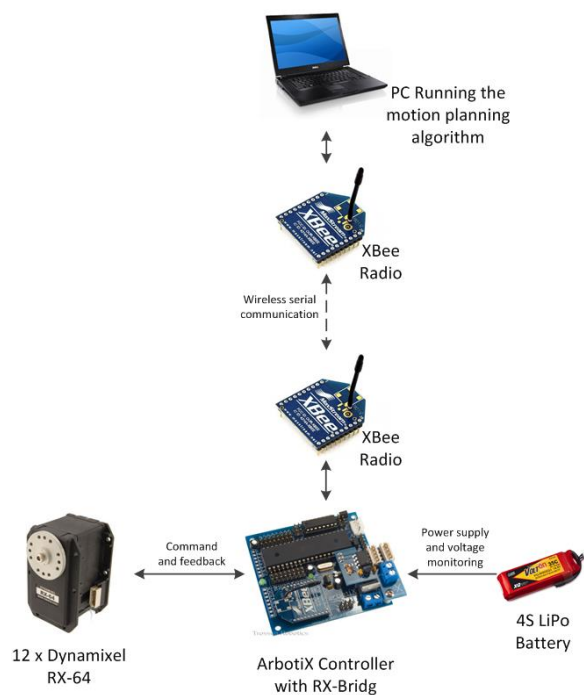
הבקר הראשי של הרובוט הוא בקר מסוג ArbotiX robocontroller [37]. בקר זה מתוכנן במיוחד למימוש תקשורת עם מנועים מסדרת Dynamixel של חברת Robotis. הבקר מבוסס על מיקרו-בקר מסוג ATMEGA644p. בבקר 32 I/O ports, תושבת לרכיב תקשורת אלחוטית מסוג XBee, דרייבר דו-ערוצי לשני מנועי DC בזרם של עד 1 אמפר ועוד. את הבקר ניתן לתכנת באמצעות סביבת Arduino IDE [38] המאפשרת פיתוח פשוט ונוח. כל 12 מנועי הרובוט מחוברים לבקר דרך RX Bridge [39] המשמש למימוש התקשורת עם המנועים. מקור המתח של הרובוט היא סוללת S LiPo4 וניתן גם להפעיל את הרובוט בחיבור חוטי לספק מתח. מתח הסוללה מנותר באמצעות הבקר כדי למנוע פריקת יתר. התקשורת עם הרובוט מתבצעת באופן אלחוטי באמצעות רכיבי תקשורת מסוג XBee 2.4Ghz [40] כאשר רכיב אחד מחובר לבקר הרובוט ושני למחשב המריץ את האלגוריתם לתכנון ההליכה.

את תוכנת הרובוט ניתן לחלק לשני חלקים. החלק הראשון הוא התוכנה הצרובה על בקר הרובוט המשמשת לפיקוד וקבלת משוב מהמנועים. החלק השני הוא אלגוריתם תכנון התנועה. החלק הראשון משתמש בספריית BioloidController [41] לצורך פיקוד מסונכרן על כל 12 המנועים. הספרייה מכילה פונקציות המאפשרות פיקוד על כל המנועים בפקודה אחת בודדת באופן מתוזמן כך שכל המנועים יגיעו לזווית היעד שנקבעה להם באותו הזמן. משך זמן התנועה הינו פרמטר הניתן לשינוי עבור כל פקודה. התכנה של הבקר קוראת פקודות המכילות 12 זוויות וזמן אינטרפולציה רצוי ומעבירה את הפקודה למנועים. למרות שבקרה מסוג זה הינה בקרת מיקום בלבד, ניתן לפי צורך עתידי לממש גם בקרת מהירות או בקרת מומנט על המנועים.

החלק השני הוא התוכנה אשר שולחת לרובוט את סט הזוויות הרצוי בכל שלבי ההליכה. ניתן לממש תוכנה זו באמצעות כל שפת תכנות התומכת במימוש תקשורת סריאלית עם רכיב ה-XBee. היתרון במבנה זה של תוכנה מאפשר למחשב בעל יכולות עיבוד חזקות לבצע את החישובים המסובכים הנדרשים כחלק מימשו האלגוריתם לתכנון התנועה אשר אינם מסוגלים להתבצע על הבקר שעל הרובוט. יתרון נוסף הוא החופש שיש למשתמש לממש את האלגוריתם במגוון שפות תכנות ללא תלות בתוכנת הרובוט. דוגמא הממחישה פשטות במימוש תוכנה כזו בסביבת MATLAB מוצגת בקוד הבא:

```
%Define the serial communication port
xbee = serial('COM27','BaudRate',38400);
%Open the serial communication port
fopen(xbee);
%Set interpolation time in milliseconds
interp_time=5000;
%Set 12 joint angles array in radians
arr=[0,pi/4,0,0,0,-pi/8,0,0,0,0,-pi/6,0];
%Convert to string before sending
joints=sprintf('%.3f,' , arr);
%Append joints and interpolation time
cmd=['!',joints,num2str(interp_time),'*'];
%Send command to the robot
fprintf(xbee,cmd);
```

כל תוכנת הרובוט מבוססת על קוד פתוח. מלבד חסכון בכסף, השימוש בקוד פתוח מאפשר שיתוף פיתוחים עתידיים על-ידי חוקרים. הפשטות במימוש השליטה ברובוט מאפשרת כתיבת דרייברים גנריים בסיבות פיתוח שונות כמו ROS [42] אשר צוברת תאוצה בשנים האחרונות. באיור 12 ניתן לראות תרשים סכמתי של התקשורת והאלקטרוניקה ברובוט.



איור 12 - תרשים סכמתי של התקשורת והאלקטרוניקה ברובוט

3.3 מידול הרובוט

כדי להגדיר באופן מוחלט את הרובוט במרחב יש להגדיר סט הפרמטרים אשר יפרשו את מרחב הקונפיגורציה של הרובוט. לצורך כך נעזר במערכות הצירים כפי שמתוארות באיור 11. המודל בו משתמש האלגוריתם לתכנון התנועה הוא מרחב קונפיגורציה בעל 18 ממדים כאשר כל ממד מציין פרמטר אחר של תנוחת הרובוט. נגדיר את תנוחת הרובוט באמצעות וקטור $q \in \mathbb{R}^{18}$ באופן הבא:

$$q = (COG, L) \quad (5)$$

כאשר $COG \in \mathbb{R}^6$ הוא וקטור המתאר את מיקום ואוריינטציית הרובוט באופן הבא:

$$COG = (C_x, C_y, C_z, \varphi_{roll}, \varphi_{pitch}, \varphi_{yaw}) \quad (6)$$

כאן C_x, C_y, C_z מגדירים את מיקום מרכז הכובד במרחב ו- $\varphi_{roll}, \varphi_{pitch}, \varphi_{yaw}$ מגדירים את זוויות הסבסוב (yaw), העלרוד (pitch) והגלגול (roll) סביב צירים x_0, y_0, z_0 בהתאמה.

הגדרה זו מגדירה את מערכת צירי גוף הרובוט (x_c, y_c, z_c) במערכת צירי העולם (x_0, y_0, z_0) כפי שניתן לראות באיור 13. כאשר התגיות COG, L ו- B מתארות את נקודות הקצה של וקטורים במערכת צירים 0. וקטור $L \in \mathbb{R}^{12}$ מתאר את מיקומי כפות הרגליים של הרובוט במערכת צירי העולם:

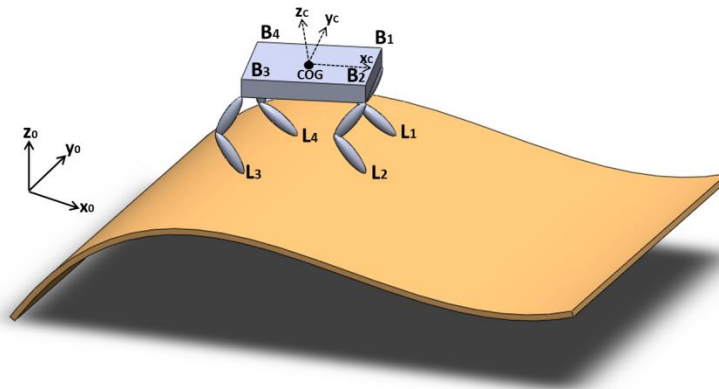
$$L = (L_1, L_2, L_3, L_4) \quad (7)$$

כאשר $L_i = (L_{ix}, L_{iy}, L_{iz})$. יש לציין שעבור משטח נתון, ה- L_{iz} מאולץ ותלוי ב- L_{ix} וב- L_{iy} .

בנוסף, נגדיר את וקטור $B \in \mathbb{R}^{12}$ אשר מגדיר את מיקומי כתפי הרובוט (נקודות חיבור הרגליים אל הגוף) במערכת צירי העולם באופן הבא:

$$B = (B_1, B_2, B_3, B_4) \quad (8)$$

כאשר $B_i = (B_{ix}, B_{iy}, B_{iz})$. וקטור זה ניתן לחישוב בהינתן וקטור q ואינו הכרחי לצורך הגדרת תנוחת הרובוט.



איור 13 - מודל הרובוט

באמצעות הגדרת מודל הרובוט ניתן לתאר את הרובוט בצורה מוחלטת במרחב. הפלט של אלגוריתם החיפוש הוא סט וקטורי q , אשר מתארים את רצף התנוחות שבהם עובר הרובוט במהלך המסלול.

3.4 קינמטיקה ישירה - Forward Kinematics

משוואות הקינמטיקה הישירה של הרובוט מאפשרת לקבל את מיקום חלקי הרובוט במרחב בהינתן זוויות המנועים. עבור סט של 12 זוויות מנועים, זווית לכל מנוע, ניתן לחשב באופן חד-ערכי את מיקומי כל הנקודות ברובוט.

על מנת לקבל את משוואות הקינמטיקה הישירה נבחרה שיטת ייצוג Zero Reference Point (ZRP) [44]. בכדי לבטא מיקום של כף רגל אחת במרחב יש לעבור תחילה ממערכת המקובעת לקרקע (מערכת 0) למערכת צירים המקובעת במרכז גוף הרובוט (מערכת C). מעבר זה ניתן לביצוע באמצעות מטריצה המבצעת הזזה למיקום מרכז הגוף C_x, C_y, C_z וסיבוב בזוויות הגוף $\varphi_{roll}, \varphi_{pitch}, \varphi_{yaw}$ על-ידי שימוש בזוויות אויילר לפי מטריצת הסיבוב הבאה [45]:

$$A_C^0 = \begin{bmatrix} c_{pitch} c_{yaw} & c_{yaw} s_{pitch} s_{roll} - c_{roll} s_{yaw} & s_{roll} s_{yaw} + c_{roll} c_{yaw} s_{pitch} & C_x \\ c_{pitch} s_{yaw} & c_{roll} c_{yaw} + c_{yaw} s_{pitch} s_{roll} & c_{roll} s_{pitch} s_{yaw} - c_{yaw} s_{roll} & C_y \\ -s_{pitch} & c_{pitch} s_{roll} & c_{pitch} c_{roll} & C_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$$s_m \triangleq \sin(\varphi_m), c_m \triangleq \cos(\varphi_m) \\ m = \{roll, pitch, yaw\}$$

כדי לעבור אל מערכת הצירים בה נמצאת דרגת החופש הראשונה של הרגל (כתף הרגל) נשתמש במטריצת ההזזה הבאה:

$$A_B^C = \begin{bmatrix} 1 & 0 & 0 & n_1 \cdot H / 2 \\ 0 & 1 & 0 & n_2 \cdot W / 2 \\ 0 & 0 & 1 & -l_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$$Leg_Index = 1 \rightarrow n_1 = n_2 = 1$$

$$Leg_Index = 2 \rightarrow n_1 = 1, n_2 = -1$$

$$Leg_Index = 3 \rightarrow n_1 = -1, n_2 = -1$$

$$Leg_Index = 4 \rightarrow n_1 = -1, n_2 = 1$$

כעת יש לכפול במטריצות הסיבוב המתאימות לדרגות החופש בכל מנוע ובמטריצות ההזזה המתאימות לאורכי החוליות המרכיבות את הרגל. הטרינספורמציות הבאות מוצגות עבור רגל בודדת אך מתאימות עבור כל אחת מהרגלים:

$$\begin{aligned}
A_1^B &= \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & -l_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
A_2^1 &= \begin{bmatrix} \cos(\theta_2) & 0 & -\sin(\theta_2) & l_2 \sin(\theta_2) \\ 0 & 1 & 0 & 0 \\ \sin(\theta_2) & 0 & \cos(\theta_2) & -l_2 \cos(\theta_2) \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
A_3^2 &= \begin{bmatrix} \cos(\theta_3) & 0 & -\sin(\theta_3) & l_3 \sin(\theta_3) \\ 0 & 1 & 0 & 0 \\ \sin(\theta_3) & 0 & \cos(\theta_3) & -l_3 \cos(\theta_3) \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{11}$$

כדי לקבל את הנוסחאות הסופיות למיקום כף הרגל יש לכפול את המטריצות הנ"ל אחת בשנייה. את התוצאה המתקבלת נכפיל בוקטור בגודל 0 כדי לקבל את מיקום הראשית של מערכת הצירים האחרונה.

$$\begin{aligned}
A_3^0 &= A_C^0 A_B^C A_1^B A_2^1 A_3^2 \\
L_i &= \begin{bmatrix} L_x \\ L_y \\ L_z \\ 1 \end{bmatrix} = A_3^0 \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
\end{aligned} \tag{12}$$

התוצאה המתקבלת מאפשרת לקבל את מיקומי כפות הרגליים של הרובוט בהינתן מיקום ואוריינטציית מרכז המסה וזוויות המנועים.

3.5 קינמטיקה הפוכה - Inverse Kinematics

משוואות הקינמטיקה ההפוכה מאפשרות לחשב עבור כל מיקום ואוריינטציה רצויים של הרובוט את זוויות המנועים המתאימות. כלומר, בהינתן מיקום רצוי לכף הרגל, משוואות הקינמטיקה ההפוכה מאפשרות לקבל את זוויות המנועים המתאימות אשר יביאו את כף הרגל למיקום הרצוי. בניגוד לקינמטיקה הישירה, לקינמטיקה ההפוכה אין פתרון חד-ערכי וקיים יותר מפתרון אחד אשר מקיים את המשוואות. המשוואות שימושיות במיוחד לאלגוריתמי תכנון צעדים המתבססים על שיטת חיפוש אשר נדרשים לחשב את זוויות מפרקי הרגל עבור מיקום אפשרי על פני הקרקע. ניתן להתייחס לרגל הרובוט כאל זרוע רובוטית הפוכה ולהשתמש בפיתוחים הקיימים בספרות [46] לקינמטיקה הפוכה של זרוע בעלת אותן דרגות חופש כמו של רגל הרובוט:

$$\begin{aligned} k &= (L_{B,x})^2 + (L_{B,y})^2 + (L_{B,z})^2 \\ s &= \sqrt{(L_{B,x})^2 + (L_{B,y})^2} \\ \theta_1 &= \text{atan}(L_{B,y} / L_{B,x}) \\ \theta_2 &= \text{atan2}(s, -L_{B,z}) \pm \text{acos}\left(\frac{k + l_2^2 - l_3^2}{2l_2\sqrt{k}}\right) \\ \theta_3 &= \pm \left(\text{acos}\left(\frac{-k + l_2^2 + l_3^2}{2l_2l_3}\right) - \pi \right) \end{aligned} \quad (13)$$

כאשר $L_{B,x}, L_{B,y}, L_{B,z}$ הם איברי הוקטור המצביע על כף הרגל במערכת צירי הכתף (מערכת B). על

מנת לקבל וקטור זה יש לחשב המטריצה אשר תעביר את הייצוג של מיקום כף הרגל במערכת העולם (מערכת 0) לייצוג במערכת הכתף (מערכת B). נציג את המטריצה המעבירה ממערכת העולם למערכת הכתף באופן הבא:

$$A_B^0 = \begin{bmatrix} \begin{bmatrix} \mathbf{R} \\ 0 \quad 0 \quad 0 \end{bmatrix} & \begin{bmatrix} d \\ 1 \end{bmatrix} \end{bmatrix} \quad (14)$$

כאשר R מייצג את הסבוב ו- d את ההזזה הקווית. נחשב את המטריצה שתבצע את המעבר ההפוך באופן הבא:

$$A_0^B = \begin{bmatrix} \begin{bmatrix} R^T \\ 0 \quad 0 \quad 0 \end{bmatrix} & \begin{bmatrix} -R^T d \\ 1 \end{bmatrix} \end{bmatrix} \quad (15)$$

כעת ניתן לחשב את הוקטור הרצוי באופן הבא :

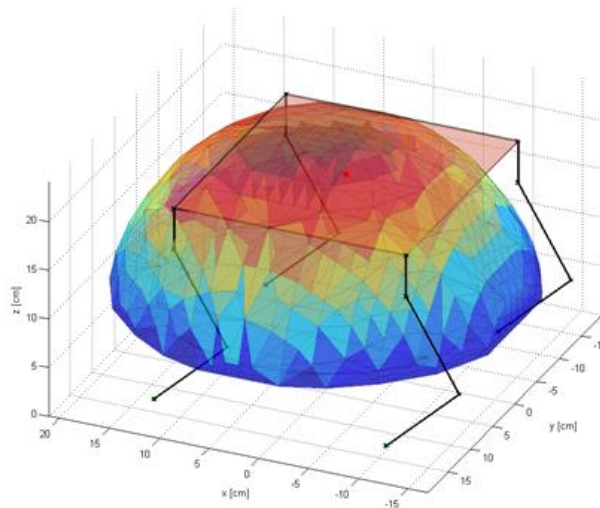
$$\begin{bmatrix} L_{B,x} \\ L_{B,y} \\ L_{B,z} \\ 1 \end{bmatrix} = A_0^B \cdot L_i \quad (16)$$

באמצעות שימוש בקינמטיקה הפוכה, האלגוריתם מסוגל להמיר את הפלט לזוויות מנועים וכך לפקוד על הרובוט לנוע. בנוסף ניתן להשתמש בקינמטיקה ההפוכה כדי לשלול צעדים אשר אינם אפשריים מבחינה קינמטית, כלומר, במידה והזוויות המחושבות מתקבלות כמספר מרוכב או כזוויות אשר מגבלות התכן של הרובוט אינן מאפשרות להגיע אליהן ניתן לשלול את המיקום הספציפי של כף הרגל אשר באמצעותו חושבו הזוויות.

3.6 מרחב העבודה של גוף הרובוט – *CKW (COG Kinematics Workspace)*

את תנועת הרובוט ניתן לחלק לשני שלבים. בשלב הראשון מתבצעת תנועה של גוף הרובוט בעוד שארבעת כפות הרגליים נשארות קבועות. בשלב השני הגוף נשאר נייח ורגלי הרובוט נעות. לצורך ניתוח מרחב התנועה בשלב הראשון של התנועה נגדיר את מרחב העבודה של גוף הרובוט.

מרחב העבודה של גוף הרובוט הוא אוסף כל הנקודות בהן יכול להימצא מרכז המסה של הרובוט בהינתן מיקומי כפות הרגליים. באופן כללי זהו מרחב 6-ממדי אך בהינתן זוויות האוריינטציה של הגוף המרחב מצטמצם ל-3 ממדים, ראה משוואה (6). את מרחב זה ניתן לחשב באופן אנליטי באמצעות משוואות הקינמטיקה ההפוכה המתארות את מיקום מרכז המסה של הרובוט אך מכיוון שהרובוט עומד על ארבעת הרגליים המערכת היא שרשרת קינמטית סגורה בעלת 12 דרגות חופש. חישוב ישיר של משוואות הקינמטיקה ההפוכה ידרוש משאבי חישוב רבים. דרך נוספת לחישוב מרחב זה היא באמצעות דיסקריטיזציה של המרחב. עבור כל נקודה אפשרית במרחב יש לבדוק האם קיים פתרון ממשי של משוואות הקינמטיקה הישירה של כל אחת מארבע הרגליים, במידה וקיים פתרון הנקודה הנבדקת שייכת למרחב ה-*CKW*. ניתן לתחום את מרחב הסריקה לממדים כאלו אשר יצמצמו את מספר הנקודות הנבדקות אך יבטיחו את כיסוי המרחב כולו. איור 14 מציג את מרחב העבודה של גוף הרובוט עבור אוריינטציות גוף אופקיות.



איור 14 - מרחב העבודה של גוף הרובוט - *CKW*

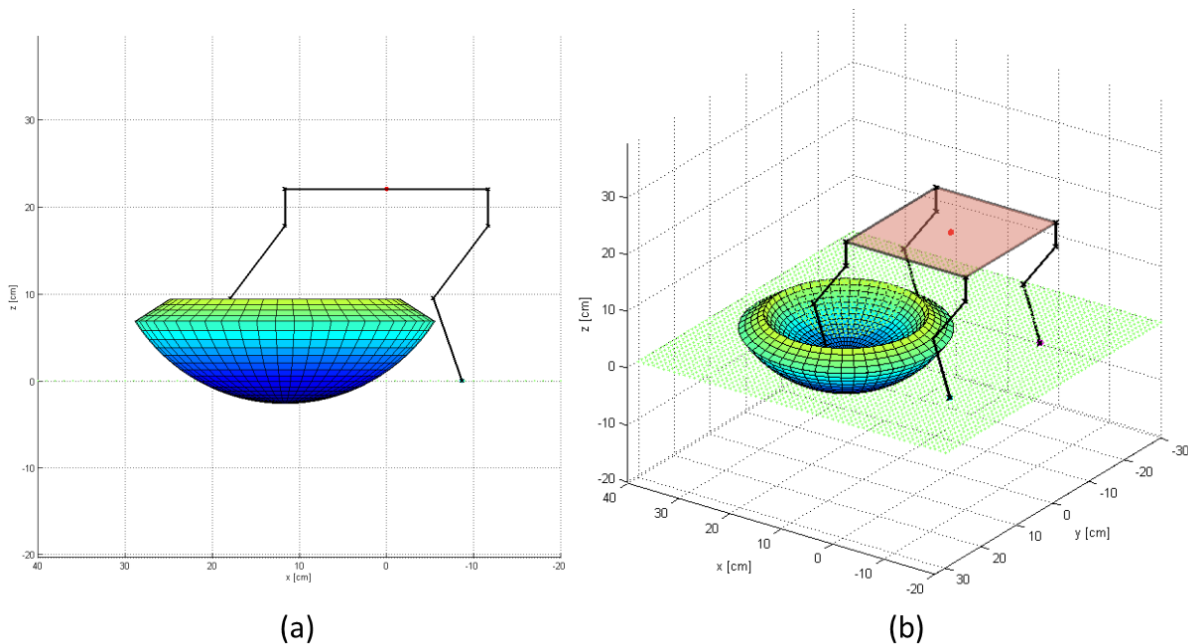
3.7 מרחב העבודה של רגלי הרובוט – LWV (Leg Workspace Volume)

חישוב מרחב העבודה של הרגל הוא שלב חשוב בעת פעולת האלגוריתם. מרחב זה מגדיר את אוסף כל המיקומים אליהם יכולה קצה הרגל להגיע מבחינה קינמטית בהינתן מיקום ואוריינטציה של גוף הרובוט. החיתוך של מרחב זה ומשטח הקרקע מאפשר לקבל את אוסף כל מיקומי הדריכה האפשריים לפני החלת האילוצים המתוארים בפרק 5. את מרחב העבודה ניתן לחשב באמצעות הקינמטיקה הישירה של הרובוט. יש לסרוק את כל הקומבינציות האפשריות של זוויות המנועים תוך שמירה על מגבלות התכן של הרובוט ועבור כל סט של שלוש זוויות לחשב את מיקום קצה הרגל. ניתן לבצע ניתוח יעיל יותר של מרחב זה באופן גאומטרי אשר דורש פחות משאבי חישוב. מכיוון ששתי דרגות החופש האחרונות של הרגל, θ_2 ו- θ_3 נמצאות על אותו מישור מרחבי ניתן לתאר את השטח על מישור זה אשר שייך למרחב העבודה של הרגל כחלק מטבעת שמרכזה הוא נקודת החיבור חולייה l_1 ו- l_2 . באמצעות טריגונומטריה פשוטה (משפט הקוסינוסים) ניתן לחשב את הרדיוס המינימלי והמקסימלי של טבעת זו:

$$\begin{aligned} R_{\max} &= l_2 + l_3 \\ R_{\min} &= \sqrt{l_2^2 + l_3^2 - 2l_2l_3 \cos(\pi - \theta_{3_Limit})} \end{aligned} \quad (17)$$

דרגת החופש במפרק θ_1 מסובבת את טבעת זו במרחב וכעת מרחב העבודה הוא חיתוך בין שני גופים מרחביים. הראשון הוא הנפח החסום בין שתי ספירות בעלות הרדיוסים שיצרו את הטבעת ומרכזן בנקודת מרכז הטבעת (עובי דופן $R_{\max} - R_{\min}$). הגוף השני הוא חרוט שיוצא מנקודת מרכז הטבעת, כיוון גובה החרוט הוא כיוון חולייה l_1 במרחב (כלפי מטה) ואורכו אינסופי. זווית המפתח של החרוט היא θ_{2_Limit} .

איור 15 מציג את מרחב העבודה של אחת מרגלי הרובוט עבור מגבלות הזוויות $\theta_{2_Limit} = 80^\circ, \theta_{3_Limit} = 80^\circ$.



איור 15 - מרחב העבודה של רגל הרובוט LWV – (a) מבט צד, (b) מבט תלת ממדי

מגבלות הזוויות הנ"ל אינן מגבלות זוויות התכן אך הן פורשות מרחב עבודה רחב מספיק על מנת לקבל חיתוך עם מישור הקרקע. החיתוך של מרחב תלת ממדי זה עם משטח הקרקע מאפשר לקבל את משטח העבודה של הרגל, ה-LWA. ניתוח מרחב העבודה באופן גיאומטרי מאפשר לקבוע האם נקודה כלשהי במרחב היא אפשרית מבחינה קינמטית. די לבדוק האם הנקודה נמצאת בתוך נפח מרחב העבודה, בניגוד לחישוב אנליטי מלא של הקינמטיקה ההפוכה. באיור 15(a) ניתן לראות שעבור תנוחה זו הרובוט מסוגל לבצע צעד בגודל מקסימלי של כ-20 ס"מ (זהו הקוטר בחיתוך מרחב העבודה עם מישור הקרקע).

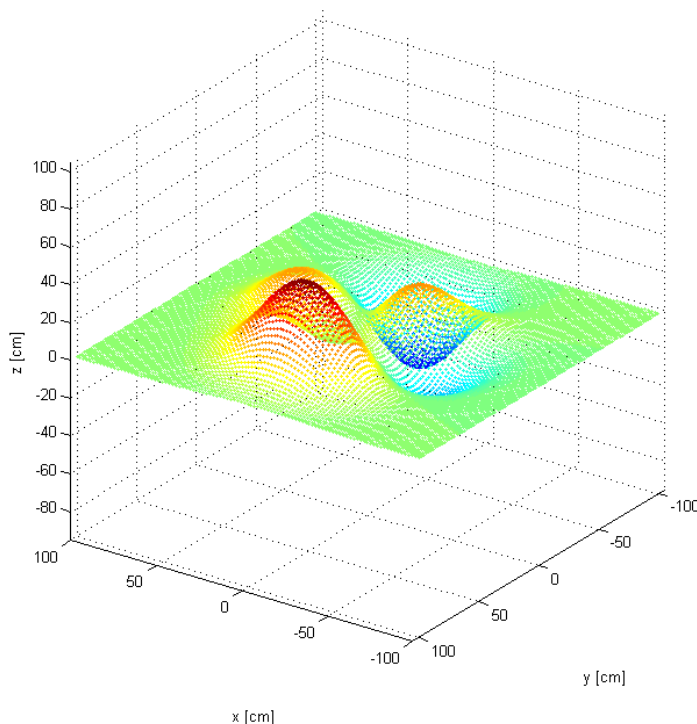
4 מידול הקרקע, ויציבות הרובוט

על מנת להתמודד עם בעיית הניווט של הרובוט יש תחילה למדל את הבעיה, כלומר, יש להגדיר את האלמנטים איתם האלגוריתם משתמש. בפרק זה אציג את האופן בו ממודלת הקרקע, את אופן המידול והגדרת מרחב הקונפיגורציה של הרובוט וכן את האופן בו מוגדר קריטריון היציבות של הרובוט. במהלך הפרק יוצגו הנחות העבודה שנלקחו במהלך פתרון הבעיה.

4.1 מודל הקרקע

מכיוון שהאלגוריתם מתבסס על שיטת חיפוש לתכנון צעדי הרובוט יש להגדיר את האופן בו מוגדר מודל הקרקע. מודל זה ישמש את האלגוריתם בשלב בדיקת היתכנות הצעדים (האם הרגל יכולה לדרוך בנקודת קרקע מסוימת) ובשלב בדיקת היציבות.

במצב אידאלי, הקרקע הייתה מאופיינת באמצעות משוואה אנליטית. במצב זה לכל נקודה במרחב קיימת פונקציה מתמטית אנליטית אשר מחזירה את גובה הקרקע בנקודה זו. בפועל, זהו אינו מצב ריאלי מכיוון שלא קיימים כיום חיישנים אשר מסוגלים לסרוק את הקרקע ולהחזיר פונקציה אנליטית. לכן, נבחרה צורת מידול דיסקרטית לפני השטח של הקרקע. לפי מודל זה, פני השטח מתוארים כגרید. רזולוציית הגריד ודיוקו תלויים בחיישן אשר באמצעותו נסרקת הקרקע. ככל שצפיפות הגריד תהיה גבוהה יותר דיוק האלגוריתם יהיה גבוה יותר וזאת עקב תיאור איכותי יותר של פני השטח והיכולת להתחשב בפרטים קטנים יותר.



איור 16 - מודל הקרקע

איור 16 מציג דוגמא למודל קרקע המבוסס על גריד בגודל 2 ס"מ, כלומר המרחק על ציר ה-y וה-x בין כל שתי נקודות שכנות הוא 2 ס"מ. השימוש במודל זה מאפשר לתאר את מיקום כף הרגל של הרובוט באמצעות שתי קואורדינטות בלבד כל עוד היא מונחת על הקרקע.

לצורך שימוש בקריטריון היציבות FER נדרש גם המידע לגבי כיוון הנורמל בנקודות הקרקע. בעיה זו ניתן לפתור באמצעות קירוב משטחים לנקודות הגריד וחישוב אנליטי של הנורמל למשטח המקורב. ניתן להשתמש בפונקציית $surfnorm$ של MATLAB לצורך ביצוע החישוב. הפונקציה מבצעת קירוב מסוג bicubic לנקודות הגריד.

4.2 קריטריון היציבות - FER (Feasible Equilibrium Region)

החישוב מתבסס על פתרון של משוואות שיווי המשקל המכילות את הכוחות והמומנטים הנוצרים מהכוחות ב- n נקודות המגע עם הקרקע

$$\sum_{i=1}^n \begin{pmatrix} f_i \\ x_i \times f_i \end{pmatrix} = - \begin{pmatrix} f_g \\ x_c \times f_g \end{pmatrix} \quad (18)$$

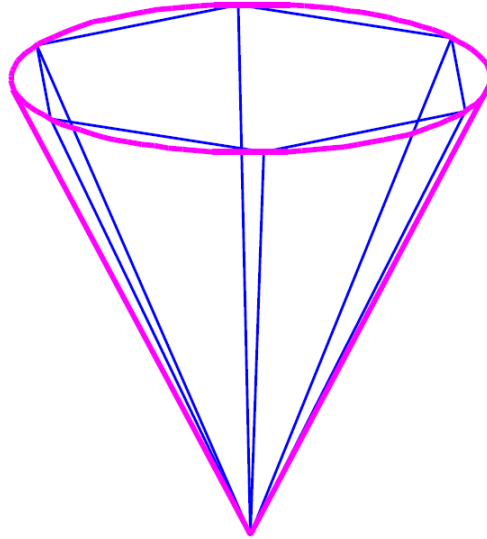
כאשר x_i הם וקטורי המיקום של נקודות המגע עם הקרקע, f_i הם וקטורי כוחות התגובה מהקרקע בנקודות המגע, f_g הוא וקטור כוח הכובד ו- $x_c = [C_x, C_y, C_z]$ הוא וקטור מיקום מרכז הכובד של הרובוט. עבור n מיקומי נקודות מגע מתקבלות $3n-4$ משוואות. למשוואות אלו מתווספות משוואות האילוצים אשר מבטיחות אי החלקה בנקודות המגע. אילוצים אלו ניתנים להמחשה כאילוץ של כיווני הכוחות בנקודות המגע להימצא בתוך מפתח קונוסי החיכוך כאשר זווית המפתח מושפעת ממקדם החיכוך.

$$\|f_i - (f_i \cdot n_i)n_i\| \leq \mu_i (f_i \cdot n_i) \quad (19)$$

כאשר n_i הוא הוקטור הנורמל לקרקע (כלפי חוץ) בנקודה x_i ו- μ_i הוא מקדם החיכוך בנקודות המגע. המשמעות של משוואה (19) היא שהגודל של הרכיב המשיקי של הכוח בנקודות המגע חייב להיות קטן או שווה ל- μ_i פעמים גודל הרכיב הנורמלי בנקודות המגע. הנעלמים במערכת המשוואות הם רכיבי הכוחות ומיקום ההיטל האופקי של מרכז המסה של הרובוט. שטח ה- FER הוא אוסף מיקומי מרכז המסה אשר מביאים למערכת עם פתרון. את אוסף המשוואות והאילוצים ניתן לתאר באמצעות סט שוויונים (משוואות שיווי המשקל) ואי-שוויונים (משוואות האילוץ). זוהי בעיה קמורה אך לא לינארית. לצורך פישוט פתרון מערכת המשוואות ניתן לבצע קירוב לינארי באמצעות מידול קונוסי החיכוך כקונוסים פוליהדרלים חסומים,

$$\begin{aligned} & (\sin((j+1)\beta) - \sin(j\beta))(f_i \cdot s_i) + (\cos(j\beta) - \cos((j+1)\beta))(f_i \cdot t_i) \\ & \leq \mu_i \sin \beta (f_i \cdot n_i), \quad j=1 \dots N \end{aligned} \quad (20)$$

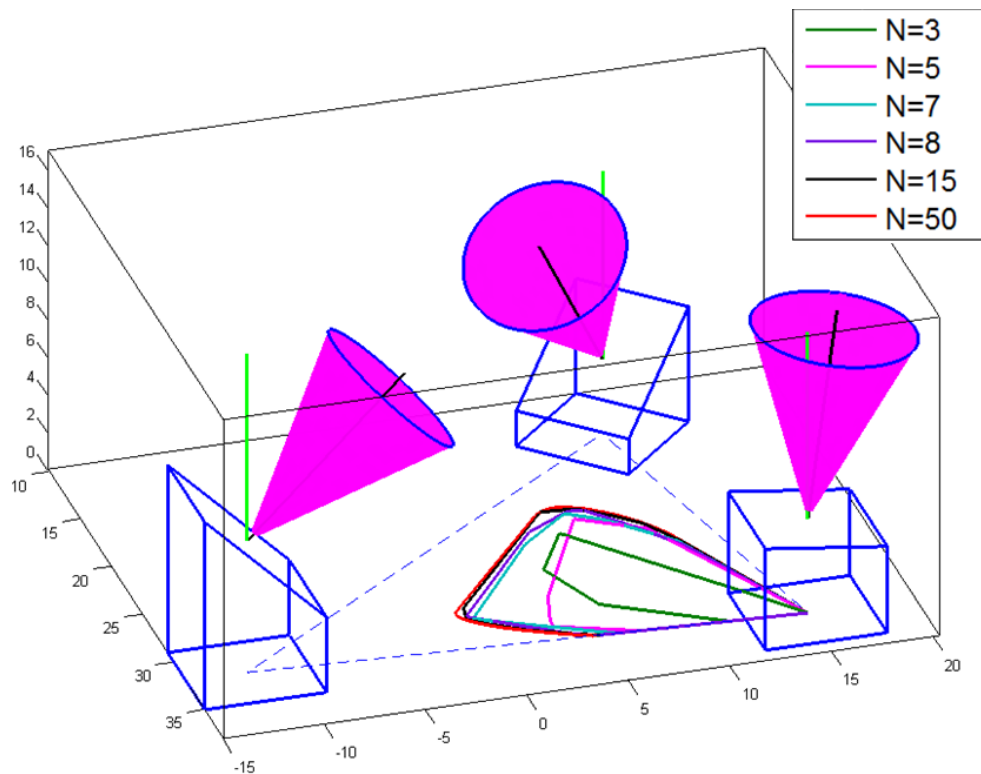
כאשר $\beta = \frac{2\pi}{N}$ ו- s_i, t_i הם וקטורי יחידה כך ש- $\{s_i, t_i, n_i\}$ היא מערכת צירים ימנית אורתוגונלית. ו- N הוא מספר הפאות.



איור 17 - קירוב קונוס החיבוכ לקונוס פוליהדרלי חסום

מידול זה המתואר באיור 17 הופך את משוואות האילוצים (19) ללינאריות ואת המערכת כולה לפוליהדרון קמור.

ככל שנשתמש ביותר פאות כך הקירוב הפנימי יהיה מדויק יותר אך עם זאת כמות המשוואות תגדל וכך סיבוכיות החישוב. קירוב זה מבטיח שהשטח המקורב יהיה מוכל בשטח האמיתי ובכך אינו "מסכן" את אמינות הקריטריון כמדד ליציבות. איור 18 מציג את תוצאות חישוב שטח ה- FER כאשר כל חישוב מתבצע עם מספר שונה של פאות.

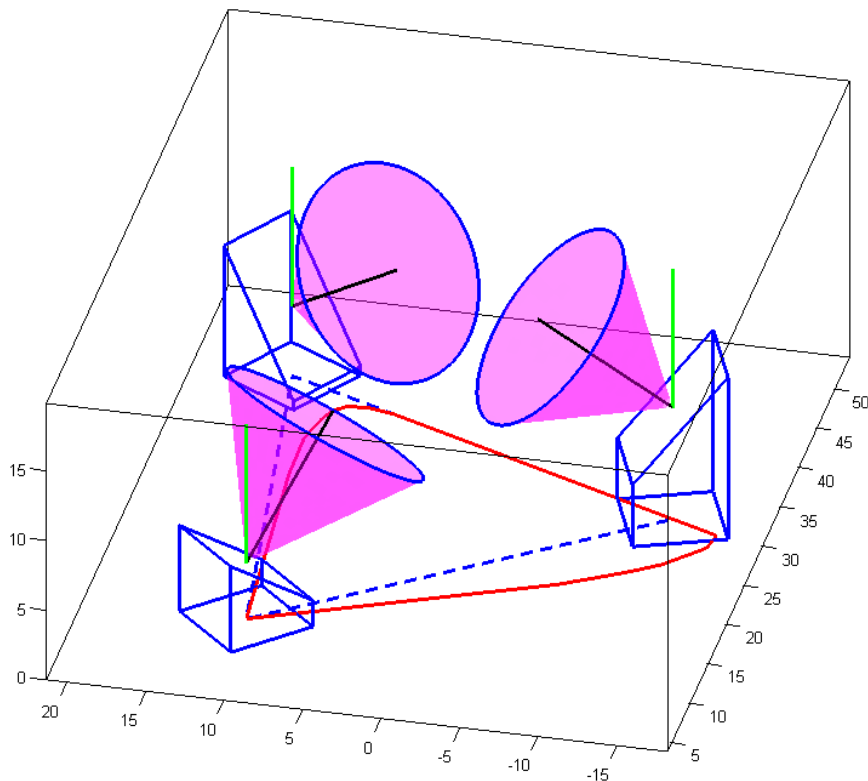


איור 18 - שטח ה- FER עבור ערכי N שונים

ניתן לראות כאשר $N=8$ מתקבל קירוב טוב מאוד לאזור המתואר באמצעות $N=50$ אשר מתאר בקירוב את האזור האמיתי.

לצורך בדיקה האם מיקום מסוים של מרכז המסה נמצא בתוך אזור ה- FER מספיק לבצע בדיקה שקיים פתרון למערכת ללא צורך לחשב את השטח כולו. לצורך חישוב השטח עצמו יש להטיל את הפוליהדרון הקמור בעל $3n-4$ ממדים על המישור הנפרש באמצעות הפרמטרים המגדירים את מיקום מרכז המסה x_c . שיטה אפשרית אחת היא שימוש בשיטת Fourier–Motzkin elimination [47] לצמצום כמות המשתנים תוך הגדלת כמות המשוואות. לאחר הצמצום ישארו רק משוואות בהן פרמטרי מיקום מרכז המסה הם הנעלמים. שיטה זו מובילה לכמות משוואות רבה מאוד עקב אופן מידול הקונוסים. שיטה נוספת ויעילה יותר היא מעבר להצגת הפוליהדרון באמצעות סט קודקודים בכל אחד מהממדים וקבלת שטח ה- FER כאוסף הקודקודים במישור פרמטרי מיקום מרכז המסה.

חשוב לציין שאזור ה- FER אינו בהכרח מוכל בתוך ה- SP (Support Polygon) ויתכנו מצבים בהם פוליון התמיכה מוכל בתוך ה- FER . איור 19 מציג דוגמא למצב כזה,



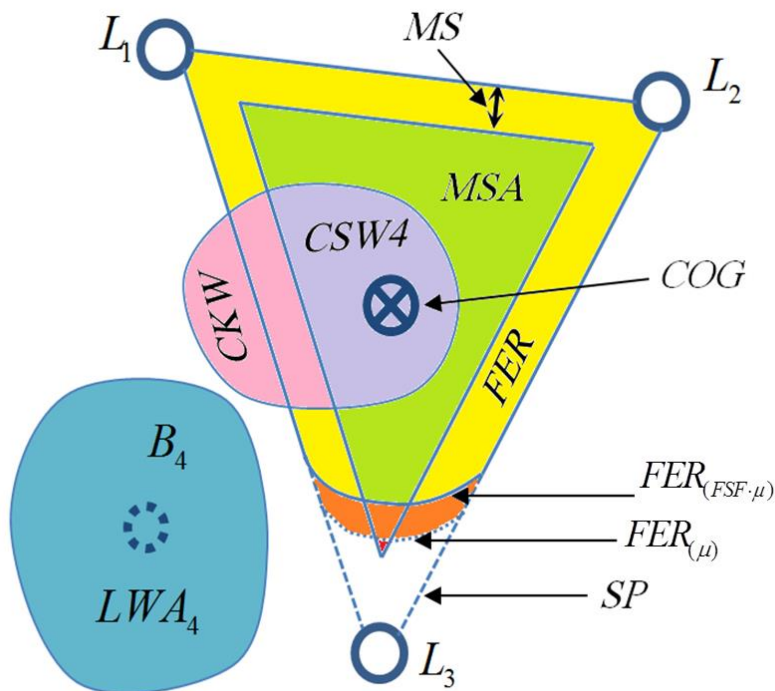
איור 19 - דוגמא לשטח FER היוצא מגבולת ה- SP

ניתוח מעמיק יותר של תכונות אזור ה- FER ניתן למצוא ב-[32]. יש לציין שקיימות גישות נוספות ושונות לפתרון הבעיה כמו למשל ב-[47] וב-[49] אשר מחשבים את גבולות האזור באופן איטרטיבי באמצעות הגרלת נקודות רנדומליות ופתרון סט בעיות אופטימיזציה קמורות. השיטה שנבחרה לשימוש באלגוריתם נמצאה כמהירה, פשוטה וקלה למימוש ובעלת יכולת התאמה למקרים בהם מספר נקודות המגע עם הקרקע אינו קבוע.

4.3 יציבות הרובוט

מרחב הקונפיגורציה של הרובוט פורש את כל הפרמטרים המגדירים את תנוחת הרובוט. לא כל התנוחות האפשריות מקיימות את אילוצי הקינמטיקה של הרובוט ואת אילוצי שווי המשקל. אוסף התנוחות החוקיות, כלומר אוסף וקטורי q אשר אפשריים למעבר הן מבחינת אילוצי הקינמטיקה והן מבחינת אילוצי שווי המשקל, מגיע לאלגוריתם וממנו נקבעת התנוחה הבאה של הרובוט.

נגדיר את פרמטר ה- ML (Moving Leg) כאינדקס של הרגל הנוכחית שמבצעת תנועה (1,2,3,4). באיור 20 ניתן לראות דוגמא להיטל של מיקומי כפות הרגליים על המישור האופקי כאשר הרגל הבאה לדריכה היא רגל 4.



איור 20 - היטל אנכי של פרמטרי הרובוט עבור $ML=4$

האיור ממחיש את אופן הגדרת קריטריון היציבות וכן את המדד ליציבות. מכיוון שהרגל הבאה לדריכה היא רגל 4, פוליגון התמיכה SP נפרש בין שלושת הרגליים האחרות. ניתן לראות ששטח ה- FER אינו מתלכד עם שטח ה- SP עקב שיפוע גדול ו/או מקדם חיכוך נמוך בנקודת המגע של רגל 3. נגדיר את ה- $MSA_i \in \mathbb{R}^2$ (Margin of Stability Area) כשטח אשר מכיל את אוסף המיקומים המקיימים את אילוצי שיווי המשקל עם מקדמי בטיחות MS (Minimum SSM) ו- FSF (FER Safety Factor) המאפשר הנעה של רגל i . מקדם ה- MS הוא המרחק משפת פוליגון התמיכה ומקדם ה- FSF הוא מספר בין 0 ל-1 אשר מקטין את המקדם החיכוך בו נעשה שימוש בחישובי ה- FER . מקדמי הבטיחות מבטיחים שמרכז הכובד לא ימוקם על שפת אזור ה- FER . מקדמים אלו הם חלק מפרמטרי הכניסה לאלגוריתם. חשוב לציין ששטח ה- MSA אינו מכיל את השטחים בהם אזור ה- FER יוצא מחוץ ל- SP (ראה איור 19).

שטחים אלו מקיימים את תנאי שיווי המשקל אך אינם יציבים דינמית. למעשה, ה- MSA הינו חיתוך של שני קריטריוני היציבות, ה- FER וה- SP .

לפני הרמת רגל מסוימת יש תחילה להזיז את מרכז הכובד אל המיקום שיבטיח את הרמת הרגל באופן יציב, כלומר, יש לקבוע את וקטור ה- COG . על מנת לאפשר לאלגוריתם החיפוש לבחור את הצעד המתאים ביותר יש לספק לו את אוסף המיקומים האפשריים. נגדיר את מרחב העבודה הקינמטי של מרכז המסה, $CKW \in \mathbb{R}^3$ (COG Kinematics Workspace) כאוסף מיקומי מרכז הכובד המקיימים את אילוצי הקינמטיקה של הרובוט. חישוב המרחב מתבצע באופן הבא:

$$CKW = f_{CKW}(L, \varphi_{roll}, \varphi_{pitch}, \varphi_{yaw}) \quad (21)$$

כאשר f_{CKW} היא פונקציית הקינמטיקה ההפוכה של נקודת מרכז הכובד של הרובוט. פונקציה זו מחשבת את כל המיקומים האפשריים של מרכז הכובד עבור זוויות גוף נתונות ומיקומי כפות רגליים נתונים. דוגמא הממחישה את קשר זה היא השוני בין מרחב העבודה של מרכז הכובד כאשר הרגליים פרושות ורחוקות מהגוף ואחת מהשנייה מתקבל מרחב קטן וכאשר הרגליים קרובות לגוף מתקבל מרחב גדול.

כדי לקבל את המיקומים מתוך מרחב העבודה הקינמטי אשר עומדים בקריטריון היציבות נגדיר את ה- $CSW_i \in \mathbb{R}^3$ (COG Stable Workspace) כאוסף הנקודות מתוך ה- CKW העומדות בקריטריון היציבות כאשר רגל i היא הרגל הנעה. אוסף נקודות אלו הן הנקודות ב- CKW אשר ההיטל האנכי שלהן נמצא בתוך שטח ה- MSA_i (ראה איור 20). חשוב לציין שלכל נקודה במרחב ה- CSW מתאימות זוויות אוריינטציה גוף ספציפיות $(\varphi_{roll}, \varphi_{pitch}, \varphi_{yaw})$, כלומר, בחירה בנקודה ממרחב זה מגדירה וקטור COG ספציפי.

ההגדרות הנ"ל מתאימות גם לשלב תנועת הגוף בעת עמידה על ארבע רגליים. בשלב זה אין חשיבות לרגל הנעה ויש ועל האלגוריתם להבטיח שמיקום מרכז הכובד יקיים את קריטריון היציבות בעת עמידה על 4 רגליים. במצב זה ה- SP הוא בעל ארבע צלעות וה- FER מחושב עבור ארבע נקודות מגע.

נגדיר את ה- $LWV_i \in \mathbb{R}^3$ (Leg Workspace Volume) כמרחב העבודה של רגל i . מרחב זה מגדיר את אוסף המיקומים במרחב התלת ממדי האפשריים מבחינה קינמטית להימצאות של כף הרגל של רגל i . מרחב זה ניתן לחישוב בהינתן מיקום ואוריינטציה (וקטור COG) של גוף הרובוט אשר מגדירים את מיקום הנקודה B_i במרחב. ומנקודה זו, באמצעות משוואות הקינמטיקה הישירה של רגל הרובוט ומגבלות התנועה של כל ציר ניתן לחשב את מרחב העבודה של הרגל.

על מנת לחשב את סט המיקומים אפשריים לדריכה של רגל מסוימת נגדיר את ה- $LWA_i \in \mathbb{R}^2$ (Leg Workspace Area) כחיתוך של ה- LWV עם משטח הקרקע. כל נקודה ב- LWA מגדירה מיקום אפשרי לדריכה מבחינה קינמטית. במידה ונחשב את ה- LWA_{ML} עבור נקודה כלשהי ב- CSW , ניתן להגיד שכל נקודה במרחב ה- LWA שחושב היא נקודת דריכה אפשרית (עבור רגל ML) שמבטיחה את יציבות הרובוט עבור אותו וקטור COG שנבחר ממרחב ה- CSW . הסיבה לכך היא שכל הנקודות במרחב ה- CSW הן

נקודות המקיימות את קריטריון היציבות של הרובוט עבור עמידה על שלושת הרגליים האחרות (שאינן רגל ML). היות והעמידה על שלוש רגליים הינה יציבה, ניתן להניח את רגל ML בכל נקודה. דרך אפשרית לחישוב מרחב ה- LWA היא בחינה של נקודות גריד הקרקע בקרבת הרובוט למשל אזור ברדיוס גודל הצעד המקסימלי הניתן לביצוע) ובדיקה האם קיים פתרון קינמטי עבורן.

מכיוון ששימוש בקריטריון ה- FER מצריך יותר משאבי חישוב האלגוריתם מבצע בדיקה ראשונית לשיפועי הקרקע בנקודות המגע ובמידה והקרקע שטוחה (זווית הנורמל לקרקע ביחס לאופק גדולה מערך סף כלשהו שניתן לקביעה) האלגוריתם משתמש בקריטריון ה- SSM בלבד תוך שימוש במקדם ה- MS .

5 אלגוריתם לתכנון הליכה

ניתן לחלק את האלגוריתם לשני שלבים עיקריים, השלב הראשון הוא שלב תכנון המסלול הראשוני. בסוף שלב זה מתקבל מסלול אשר ינווט את הרובוט עד לנקודת היעד. מסלול זה אינו בהכרח המסלול האופטימלי אך במידה וקיים מסלול אשר ניתן למעבר על ידי הרובוט ובהנחה שרזולוציית הגריד אינה "מחסירה" מידע הכרחי – ימצא מסלול. השלב הראשון כולל את שלב הבנייה והחיפוש בגרף באמצעות פונקציית הפוטנציאל. השלב השני של האלגוריתם כולל עיבוד נוסף (Post-Processing) אשר תפקידו להתמודד עם בעיית נקודות המינימום המקומיות ולשפר את מסלול הניווט שהתקבל בשלב הראשון.

כפי שתואר בפרק 2.2, הליכת הרובוט מורכבת מרצף של שתי הזזות רגלים, הזזת גוף ושוב שתי הזזות רגליים וכך הלאה. אופן הליכה זה נבחר מכיוון שברוב במקרים הוא מאפשר הליכה בה גוף הרובוט נע בכיוון התקדמות הרובוט באופן מגמתי, כלומר, כיוון התנועה של גוף הרובוט יהיה לעבר כיוון ההתקדמות הכללי של הרובוט. בתנאי שטח קשים במיוחד יתכנו תנועות שאינן לעבר היעד. חשוב לציין שאופן הליכה זה מכיל בתוכו את האפשרות לבצע הליכה באופן בו אחרי כל תנועת רגל מתבצעת תנועת גוף שכן זהו מקרה פרטי בו הרגל השנייה מבין שתי הרגליים הנעות דורכת במקום.

נגדיר את פרמטר ה- NML (Next Moving Leg) כאינדקס של הרגל השנייה להזזה ברצף הזזת הרגלים הנוכחי. כפי שניתן לראות במשוואה (21), על מנת לחשב את CKW יש לדעת את מיקומי כפות הרגליים הנוכחיים (וקטור L). מכאן ניתן להסיק שחישוב ה- LWA_{NML} תלוי בבחירת וקטור L_{ML} מסוים בשלב הצעד הראשון ברצף. כלומר, עבור ML ו- NML נתונים, לא ניתן לחשב את מיקומי הדריכה אפשריים של הרגל השנייה ברצף ללא קביעת מיקום דריכת הרגל הראשונה. עובדה זו מביאה למסקנה שבכדי לקבל את אוסף כל אפשרויות התנועה עבור צעד כפול של שתי רגליים מסוימות יש לחשב את מרחב ה- CSW_{ML} . עבור כל נקודה במרחב (וקטור COG) יש לחשב את מרחב ה- LWA_{ML} ועבור כל נקודה במרחב זה (L_{ML}) יש לחשב את ה- LWA_{NML} המתאים לה ומתוכו לבחור את ה- L_{NML} הרצוי. בחירה זו משמעותה בחירה ב- COG, L_{ML}, L_{NML} אשר יגדירו את הצעד הכפול הבא.

על מנת להניע את הרובוט לעבר נקודת היעד יש להגדיר את הנקודה ברובוט אותה יש לנווט, BP (Body Position). זוהי הנקודה אשר בסופו של דבר תגיע לנקודת היעד. נגדיר נקודה זו באופן הבא:

$$BP = f_{BP}(q) \quad (22)$$

כאשר ניתן להגדיר את f_{BP} בכל דרך רצויה ויש להזין אותה כאחד מנתוני הקלט לאלגוריתם. לדוגמא, ניתן להגדיר את פונקציה זו כמיקום הממוצע של כל כפות רגלי הרובוט:

$$BP = \frac{1}{4} \left[\sum_{i=1}^4 L_{ix}, \sum_{i=1}^4 L_{iy}, \sum_{i=1}^4 L_{iz} \right] \quad (23)$$

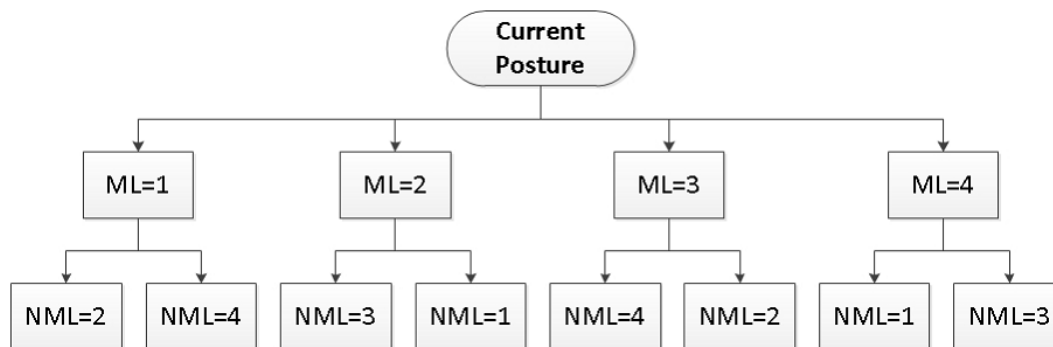
או כמיקום מרכז המסה:

$$BP = [C_x, C_y, C_z] \quad (24)$$

כעת נגדיר את אופן בניית הגרף בו מתבצע החיפוש. כל צומת בגרף מייצגת וקטור תנוחת רובוט (q) ספציפי. קשת בגרף מתארת מעבר אפשרי בין שתי תנוחות עוקבות, כלומר, קיום אילוצי הקינמטיקה ואילוצי שיווי המשקל במעבר בין התנוחות. כדי למצוא מסלול מתנוחה התחלתית כלשהי אל תנוחה סופית רצויה, האלגוריתם משתמש בשיטת החיפוש DFS כדי לסרוק את כל הצעדים הכפולים האפשריים. כאשר צעד בודד אפשרי מתואר באמצעות כל הצמתים המחוברים לצומת הנוכחי באמצעות קשת. לכל צעד כפול אפשרי מחושב הפוטנציאל ובאמצעות השוואה בין כל ערכי הפוטנציאל, נבחר הצעד הכפול בעל הפוטנציאל הנמוך ביותר. חשוב לציין שבמידה וצעד בודד ולא צעד כפול הוא דרך ההתקדמות הטובה ביותר אז פתרון זה יבחר באמצעות בחירת צעד כפול שבו הרגל הראשונה דורכת במקום.

את פונקציית הפוטנציאל U , ניתן להגדיר בצורות שונות כדי להביא לתנועת רובוט בעלת מאפייני תנועה רצויים. לדוגמא, במידה ופונקציית הפוטנציאל תוגדר כמרחק של ה BP מנקודת היעד, יבחרו צירופי הצעדים אשר יקרבו את הרובוט ליעד באופן הטוב ביותר. במידה ורצוי ניווט לאורך מסלול נתון, ניתן להוסיף לפונקציית הפוטנציאל גם איבר שערכו הוא המרחק מהמסלול וכך הרובוט ישאף להתקרב למסלול בנוסף להתקרבות לנקודת היעד. ניתן להוסיף איברים נוספים שיביאו למאפייני תנועה רצויים כגון שמירה על חזית הרובוט כלפי היעד או איברים אשר יצמצמו את צריכת האנרגיה במהלך התנועה. כל איבר המרכיב את פונקציית הפוטנציאל מוכפל במשקל מסוים אשר מכתוב את רמת ההשפעה שלו ביחס לאיברים האחרים. באמצעות נרמול איברי הפונקציה וחלוקה בסכום המשקלים ניתן לנרמל את ערך הפוטנציאל.

כדי לקחת בחשבון את כל כיווני ההתקדמות האפשריים יש לבצע חיפוש בכל אפשרויות ההתקדמות עבור כל קומבינציית רגליים. כמובן שצעד כפול באמצעות שתי רגליים אלכסוניות אינו אפשרי מכיוון שלא קיים מיקום של מרכז הכובד שיאפשר תנועה זו באופן יציב. במקרה זה על מרכז הכובד להיות מעל הקו המחבר את שתי הרגליים שאינן מבצעות צעד ולא ניתן להבטיח זאת בדיוק. איור 21 מציג את כל הקומבינציות האפשריות שיש לבדוק עבור כל חיפוש של צעד כפול כאשר ML (Moving Leg) הוא אינדקס הרגל המבצעת את הצעת הראשון בצעד הכפול ו- NML (Next Moving Leg) הוא אינדקס הרגל השנייה.



איור 21 - שמונת הקומבינציות האפשריות לצעד כפול

לאחר בחירת הצעד הכפול בעל הפוטנציאל הנמוך ביותר בכל אחת משמונת הקומבינציות יש להשוות בין שמונת הפוטנציאלים ולבחור את הקומבינציה שהביאה לצעד הכפול בעל הפוטנציאל הנמוך ביותר.

ניתן להציג את הבעיה שיש לפתור בכל חיפוש כבעיית אופטימיזציה באופן הבא :

$$\begin{aligned}
 q_j &= \arg \min(U_j(q_j)) \\
 s.t. \\
 COG &= (C_x, C_y, C_z, \varphi_{roll}, \varphi_{pitch}, \varphi_{yaw}) \in CSW_{ML_j}, \\
 L_{ML_j} &\in LWA_{ML_j}, \\
 L_{NML_j} &\in LWA_{NML_j}.
 \end{aligned} \tag{25}$$

כאשר U_j היא פונקציית הפוטנציאל ו- j הוא האינדקס שמציין את קומבינציית הרגליים הנוכחית ($j=1,2,\dots,8$). פתרון של בעיה זה עבור כל j ובחירת הקומבינציה עם הפוטנציאל הנמוך ביותר תביא לבחירת הצעד הכפול הבא, $U_{\min} = \min(U_1, U_2, \dots, U_8)$.

באופן תאורטי האלגוריתם צריך לרוץ עד שערך הפוטנציאל מגיע לאפס. בפועל, בעקבות הרזולוציה של מידול הקרקע ורזולוציית החיפוש וכן עקב הרצון להימנע מהמון צעדים קטנים שיביאו להתכנסות מדויקת, יש להגדיר ערך סף מינימלי (LPT (Low Potential Threshold), אשר כאשר ערך הפוטנציאל יורד מתחת סף זה הוא מניח שהמטרה הושגה והניווט הסתיים.

כדי להתמודד עם בעיית המינימום המקומי והגעה לנקודת בהם הרובוט נתקע (Deadlock) ואינו "רוצה" להתקדם וכן כדי להימנע ממצבים לא רצויים כגון הצלבת רגליים או אי היכולת להתקדם בכל כיוון יש להחיל אילוצים נוספים על אופן בחירת הצעדים :

אילוץ A – צעד אשר מביא להצלבת רגליים (מצב בו ההיטל האופקי של הרגל הדורכת נמצא בתוך ההיטל האופקי של המשולש המוגדר באמצעות שלושת כפות הרגליים התומכות) הוא אינו צעד "חוקי" ולא יכנס לאוסף הצעדים האפשריים. בנוסף, צעד המביא למצב בו זוג כפות רגליים צמודות זו לזו (במרחק בין כפות הרגליים קטן ממרחק הנקבע מראש) אינו צעד "חוקי". אילוץ זה מקטין את הסבירות להתנגשות בין רגליים.

אילוץ B – מיקום הצעד הראשון (L_{ML}) חייב לאפשר שטח גדול מאפס עבור ה- MSA_{NML} . אילוץ זה מבטיח שהרובוט יישאר יציב בעת הרמת הרגל בשנייה בצעד הכפול.

אילוץ C – מיקום הצעד השני (L_{NML}) חייב להביא למצב בו כל אחד ממרחבי ה- CSW_{ML} של כל אחת מהרגלים הבאות אינו ריק. אילוץ זה מבטיח שיהיה אפשר לבצע את הצעד הבא באמצעות כל אחת מהרגליים. האילוץ מגדיל את האפשרות לתנועה רב-כיוונית של הרובוט. אילוץ זה ניתן לביטול הן לצרכי צמצום זמני החישוב והן בקרים בהם תנאי השטח אינם מורכבים וההתקדמות בכיוון אחד באופן מגמתי היא אפשרית.

אילוץ D – למיקום הצעד השני (L_{NML}) אסור להיבחר להיות וקטור L אשר נבחר בעבר (או בפועל קרוב מאוד). אילוץ זה ניתן לכיוון באמצעות שינוי הגדרות הזהות בין הוקטורים. אילוץ זה מהווה את זיכרון אלגוריתם חיפוש ה- DFS ומונע דריכה במקום או חזרה לאותה נקודה ובכך מאפשר התמודדות עם בעיית המינימום המקומי. המניעה מהשארות באותה תנוחה מחייבת את הרובוט לנוע למרות שערכי הפוטנציאל יעלו מעבר לערך הנוכחי. ניתן לדמות את פעולת אילוץ זה כמילוי ה"שקע" של המינימום המקומי בזמן.

במידה ומתקבל מצב בו לא קיימים מיקומי דריכה אפשריים כלל, ניתן להסיק שלא קיים מסלול תנועה אפשרי וזאת עקב העובדה שכל צעדי הרובוט האפשריים נשללו. אי קיום המסלול יכול לנבוע מתנאי שטח שאינם עבירים או עקב מידול אשר בו רזולוציית גריד הקרקע או רזולוציית החיפוש מחסירות מידע, למשל כאשר רווחי הגריד גדולים מידי.

הפסאודו-קוד הבא מתאר את השלב הראשון של האלגוריתם:

Algorithm 1: Algorithm first stage

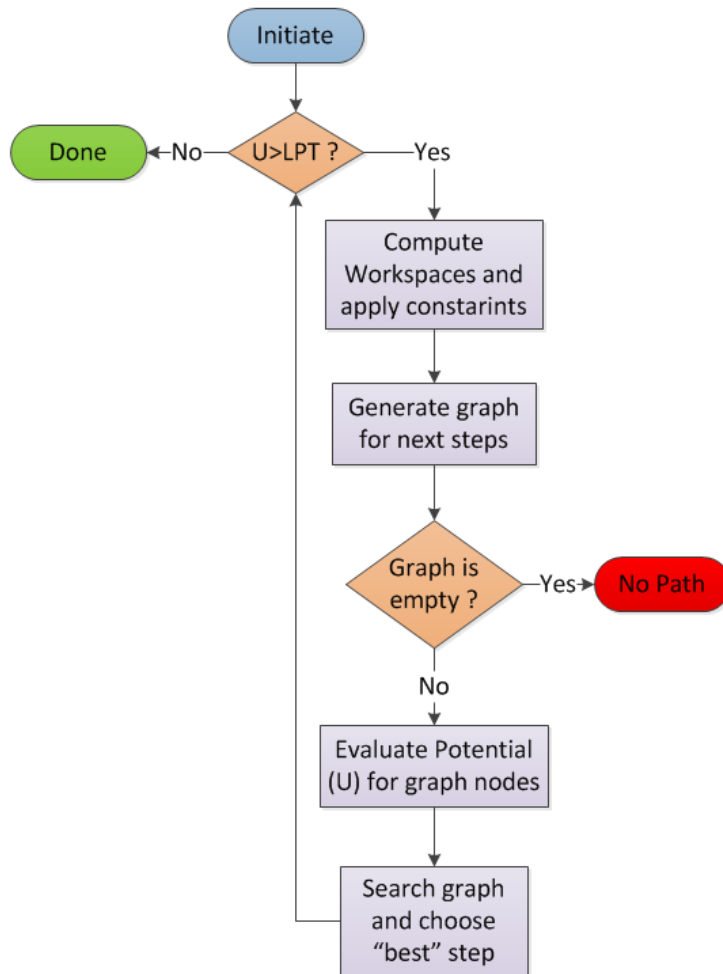
input : Robot kinematics, APF function and weights, $f_{BP}(q)$, q_0 , q_{goal} , LPT , MS , FSF , μ
output: A sequence of navigable stable postures

```

1 while  $U > LPT$  do
2   Calculate  $CKW$ ;
3   foreach Possible two steps sequence ( $j=1...8$ ) do
4     Calculate  $FER \xrightarrow{MS,FSF} MSA$ ;
5      $CSW_{ML_j} = CKW \cap MSA_{ML_j}$ ;
6     foreach  $COG_{jk}$  in  $CSW_{ML_j}$  ( $k=1,2...$ ) do
7        $LWV_{ML_{jk}} = f_{LWV}(COG_{jk} \in CSW_{ML_j})$ ;
8        $LWA_{ML_{jk}} = LWV_{ML_{jk}} \cap Ground$ ;
9       Apply constraints A,B and D over  $LWA_{ML_{jk}}$ ;
10      foreach  $L_{ML_{jkn}}$  in  $LWA_{ML_{jk}}$  ( $n=1,2...$ ) do
11         $LWV_{NML_{jkn}} = f_{LWV}(L_{ML_{jkn}} \in LWA_{ML_{jk}})$ ;
12         $LWAN_{ML_{jkn}} = LWV_{NML_{jkn}} \cap Ground$ ;
13        Apply constraints A,C and D over  $LWAN_{ML_{jkn}}$ ;
14        foreach  $L_{NML_{jkni}}$  in  $LWAN_{ML_{jkn}}$  ( $i=1,2...$ ) do
15           $U_{jkni} = f(COG_{jk}, L_{ML_{jkn}}, L_{NML_{jkni}})$ 
16      if A possible step exist ( $i > 0$ ) then
17         $U_{min} = \min(U_{jkni})$ , for all  $j,k,n,i$ ;
18        Define current posture:  $q = \arg(U_{min})$ ;
19        Log  $q$ ;
20      else
21        DFS eliminate current  $q$ ;
22        Current  $q$  = previous  $q$ ;
23      if current  $q=q_0$  then Deadlock;

```

תיאור כללי של האלגוריתם באמצעות תרשים זרימה מופיע באיור 22.



איור 22 - תרשים זרימה מפורט של האלגוריתם

תפקידו של השלב השני של האלגוריתם הוא לשפר את תוצאת הפלט של השלב הראשון במידה ויש צורך. בשלב זה מתבצע זיהוי של נקודת היציאה מהמינימום המקומי ונקבעת נקודת ביניים כנקודת יעד ביניים במסלול. לאחר קביעת נקודת הביניים, יש להריץ את השלב הראשון של האלגוריתם כאשר כעת המסלול לנקודת היעד מאולץ לעבור דרך נקודת הביניים. במצב זה הסיכוי להיכנס אל המינימום המקומי קטן יותר. ניתן להריץ את השלב השני פעם נוספת על הפתרון שהתקבל עד לקבלת מסלול סופי רצוי. כמובן שהרצת השלב השני אינה מתאימה לניווט הרבובט בזמן אמת.

אחד היתרונות המרכזיים של האלגוריתם היא ההתייחסות לכל נקודות גריד הקרקע כאל שוות. כלומר, האלגוריתם אינו מסווג אזורים מסוימים כאסורים לדריכה. זהו יתרון מכיוון שבאופן פעולה זה גם באזורים אשר בעיני אדם המתבונן אליהם יראו כקשים/מסובכים למעבר האלגוריתם ינסה לתכנן מסלול תנועה (כמובן בתנאי שהאזורים נמצאים בדרך הקצרה אל היעד). תופעה זו מומחשת בבירור בתוצאות הסימולציות והניסויים בהם המסלול המתקבל מהאלגוריתם מאלץ את הרבובט לדרך בקרקע משופעת

ולא לעקוף את מכשולי קרקע. נקודות הדריכה הנבחרות על גבי הקרקע המשופעת מקיימות את אילוצי שיווי המשקל של הרובוט ולכן מובטח שהרובוט לא ייפול במהלך התנועה.

משפט: האלגוריתם שלם במובן שאם קיים מסלול על הגריד, הוא ימצא ואם לא קיים מסלול האלגוריתם ייעצר בסמן סופי וידווח שלא קיים מסלול.

הוכחה: הגרף אשר נבנה בעת עבודת האלגוריתם מתאר את כל המסלולים האפשריים ולכן עבור גרף בגודל סופי אלגוריתם ה-DFS הוא אלגוריתם שלם [50]. האלגוריתם מבוסס על מימוש אלגוריתם ה-DFS ולכן גם הוא שלם. חשוב לציין שתאור כל המסלולים האפשריים מותנה בבחירת רזולוציית גריד אשר אינה מחסירה מידע וכן בחירה של רזולוציה מתאימה עבור מרחבי העבודה של הרובוט. כלומר, ישקלו כל אפשרויות התנועה של הרובוט עבור כל צעד. במקרה הגרוע ביותר האלגוריתם יסרוק את כל אפשרויות התנועה האפשריות (כל הגרף) ולאחר מכן ידווח שלא נמצא מסלול.

משפט: סיבוכיות זמן הריצה של האלגוריתם הינה מסדר $O(m^2 n^2 k^4)$. כאשר מרחב העבודה של רגל הרובוט (LWA) בגודל n נקודות, מרחב העבודה של גוף הרובוט (CKW) בגודל m נקודות וגריד הקרקע בעל k נקודות.

הוכחה: חישוב ה-CSW הוא חישוב מסדר $O(4m)$, חישוב ה- LWA_{ML} הוא מסדר $O(n)$, חישוב ה- LWA_{NML} הוא מסדר $O(2n)$ (שתי אפשרויות לרגל שנייה בצעד), החלת אילוץ C היא מסדר $O(2m)$ וחיפוש ה-DFS הוא מסדר $O\left(\frac{k!}{(k-4)!}\right) \approx O(k^4)$. סיבוכיות חיפוש ה-DFS מחושבת לפי החסם העליון שמוגדר לפי כמות האפשרויות למקם את 4 רגלי הרובוט על הגריד. זהו חסם עליון מכיוון שקיימים אילוצים קינמטיים אשר מצמצמים את כמות האפשרויות. התוצאה המתקבלת היא אלגוריתם מסדר $O(m^2 n^2 k^4)$. זוהי סיבוכיות פולינומלית.

ללא החלת אילוץ C (אשר מבטיח שיהיה אפשר לבצע את הצעד הבא באמצעות כל אחת מהרגליים) הסיבוכיות יורדת לסדר $O(mn^2 k^4)$. כאשר מרחב ה-LWA מחושב לפי בחירת הנקודות מגריד הקרקע המקיימות את האילוצים הקינמטיים, קיים קשר בין צפיפות גריד הקרקע לכמות הנקודות במרחב ה-LWA. המקרה בו מספר הנקודות הוא מקסימלי הוא כאשר ל-LWA צורת דיסקה בקוטר הצעד המקסימלי שיכול לבצע הרובוט ואז מתקבל הקשר $n = \frac{\pi \cdot Max_Step^2}{4 \cdot Grid_resolution^2}$. כאשר $Grid_resolution$ הוא המרווח בין שתי נקודות סמוכות בגריד הקרקע ו- Max_Step הוא גודל הצעד המקסימלי שהרובוט יכול לבצע. הקשר למספר הנקודות בגריד הוא $k = \frac{Ground_Area}{Grid_resolution^2}$ כאשר

$Ground_Area$ הוא השטח הכולל של הקרקע. מכאן ניתן להגיע לסיבוכיות של $O\left(m / Grid_resolution^{12}\right)$ הגדלים m, n ו- k , ניתנים לקביעה בעת הפעלת האלגוריתם ומאפשרים בחירה בין רזולוציה גבוהה לבין זמן ריצה נמוך. בדרך כלל מספר הנקודות m ו- n קשור אחד לשני ומושפע מהמגבלות הקינמטיות ומגבלות התכן של הרובוט.

6 סימולציות

6.1 מבנה עקרוני של התוכנה המממשת את האלגוריתם

על מנת להוכיח את היתכנות האלגוריתם ויעילותו יש לממשו בסביבה ממוחשבת לצורך ביצוע סימולציות וכן ליצור בסיס לביצוע ניסויי היתכנות. התוכנה שנבחרה למימוש האלגוריתם היא MATLAB והאלגוריתם מומש באמצעות כתיבת קוד המבצע את שלבי האלגוריתם.

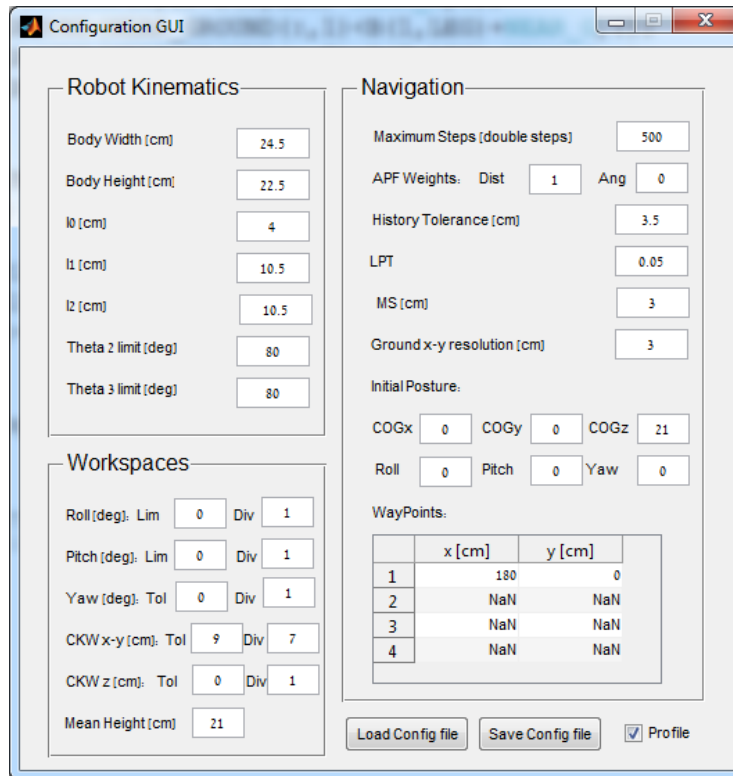
התוכנה נכתבה במבנה פונקציונלי כאשר כל פונקציה מבצעת חישוב ספציפי. בעת הרצת התוכנה מתבצעות קריאות לפונקציות המתאימות לשלב החישוב הנוכחי. מכיוון שהאלגוריתם תוכנן באופן גנרי כך שלא יהיה תלוי במבנה קינמטי של רובוט מסוים עלה הצורך לכתוב את התוכנה המממשת את האלגוריתם באופן כזה שתאפשר שינוי פשוט של פרמטרי הכניסה וההגדרות הנדרשות למימוש האלגוריתם. לצורך כך נכתבה תוכנה נוספת בעלת ממשק משתמש אשר מאפשרת טעינה ושמירה של קבצי הגדרות. בעת הרצת התוכנה המממשת את האלגוריתם המשתמש מתבקש לטעון קובץ הגדרות אשר ישמש לריצת התוכנה הנוכחית.

טבלה 2 מציגה פירוט על הפרמטרים העיקריים המשפיעים על אופן ריצת האלגוריתם.

טבלה 2 - הפרמטרים העיקריים המשפיעים על אופן ריצת האלגוריתם

<i>Grid _ resolution</i>	המרחק בין הנקודות בגריד הקרקע
<i>MS</i>	רף היציבות עבור ה- <i>SP</i> כפי שמוגדר בפרק 4.3
<i>FSF</i>	מקדם הבטיחות עבור קריטריון ה- <i>FER</i> כפי שמוגדר בפרק 4.3
<i>LPT</i>	ערך הפוטנציאל המינימלי כפי שמתואר בפרק 5
<i>Constraint_D_identity_threshold</i>	המרחק באמצעותו אילוץ <i>D</i> מגדיר וקטורי <i>L</i> זהים
<i>CKW _ resolution</i>	המרווח בין הנקודות במרחב ה- <i>CKW</i>

איור 23 מציג ממשק גרפי המשמש לצורך קביעה, שמירה וטעינה של פרמטרים אלו ופרמטרים נוספים.



איור 23 - ממשק משתמש לקביעת הגדרות

פירוט מורחב על אופן מימוש האלגוריתם באמצעות תיאור תפקידן ופעולתן של הפונקציות העיקריות המרכיבות את הקוד מופיע בנספח A.

6.2 תוצאות סימולציות

בפרק זה מוצגות תוצאות של סימולציות אשר ממחישות את אופן פעולת שלבי האלגוריתם (סימולציות I ו-II), תוצאות סימולציה להמחשת יתרון השימוש בקריטריון ה- FER (סימולציה III), תוצאות עבור תכנון תנועה בשטח מורכב (סימולציה IV) וכן ניתוח זמני ריצה (סימולציה V). פונקציית הפוטנציאל הוגדרה עבור הסימולציות באופן הבא:

$$U = \frac{w_1 \hat{U}_{Dist} + w_2 \hat{U}_{Heading} + w_3 \hat{U}_{Path}}{w_1 + w_2 + w_3} \quad (26)$$

$$w = [w_1, w_2, w_3]$$

כאשר \hat{U}_{Dist} היא הפונקציה המנרמלת את המרחק של נקודה BP אל נקודת היעד, \hat{U}_{Path} היא פונקציה המנרמלת את המרחק מהמסלול שהוגדר לניווט אל היעד. ו- $\hat{U}_{Heading}$ היא פונקציה המנרמלת את זווית הסטייה של חזית הרובוט מהכיוון אל נקודת היעד. יש לציין שאיברי הפונקציה הינם תמיד חיוביים. הפונקציה f_{BP} הוגדרה לפי משוואה (23), פונקציה זו מתארת את הנקודה אותה האלגוריתם מנווט באמצעות המיקום הממוצע של ארבעת כפות הרגליים.

6.2.1 סימולציה I

הנתונים הבאים הם ערכי הפרמטרים אשר הוזנו לאלגוריתם בהרצת הסימולציה :

$$Grid_resolution = 3[cm]$$

$$MS = 3[cm]$$

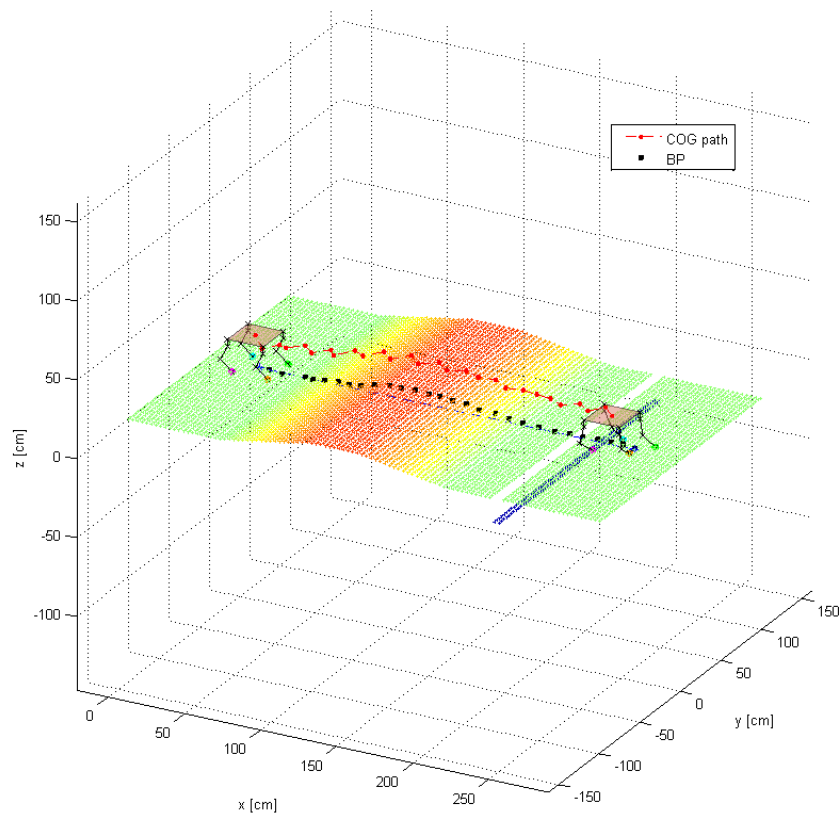
$$LPT = 0.03$$

$$Constraint_D_identity_threshold = 3.5[cm]$$

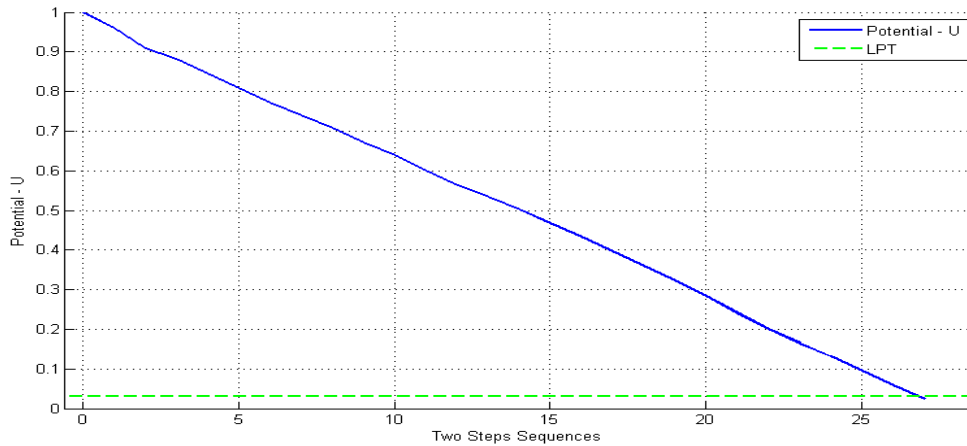
$$CKW_resolution = 2[cm]$$

$$w = [1, 0, 0]$$

הסימולציה הראשונה מציגה את פעולת האלגוריתם בסביבה ללא מכשולים אשר עלולים לגרום לבעיית מינימום מקומי. על הרובוט להגיע אל נקודת היעד תוך כדי מעבר מעל תעלה צרה ועמוקה. תוצאות הסימולציה מוצגות באיורים הבאים :



איור 24 - תוצאות סימולציה I – פלט האלגוריתם



איור 25 - תוצאות סימולציה I - שינוי הפוטנציאל

כפי שניתן לראות באיור 24, פלט האלגוריתם הוא רצף של תנוחות הרובוט אשר מביאות אותו אל נקודת היעד כאשר המסלול האדום המוצג בגרף הוא מסלול מרכז המסה של הרובוט והריבועים השחורים הם מיקומי ה-BP. באיור 25 ניתן לראות את ירידת הפוטנציאל במהלך התקדמות הרובוט עד לירידה מתחת לערך ה-LPT.

6.2.2 סימולציה II

הסימולציה השנייה מוצגת לשם המחשת הצורך והשימוש בשלב השני של האלגוריתם. כעת על הרובוט להגיע על נקודת היעד תוך הימנעות מנפילה לבור רחב ועמוק. חשוב לציין שהאלגוריתם אינו "מודע" להימצאות הבור עד לרגע ההיתקלות בו. הנתונים הבאים הם ערכי הפרמטרים אשר הוזנו לאלגוריתם בהרצת הסימולציה:

$$Grid_resolution = 4[cm]$$

$$MS = 3[cm]$$

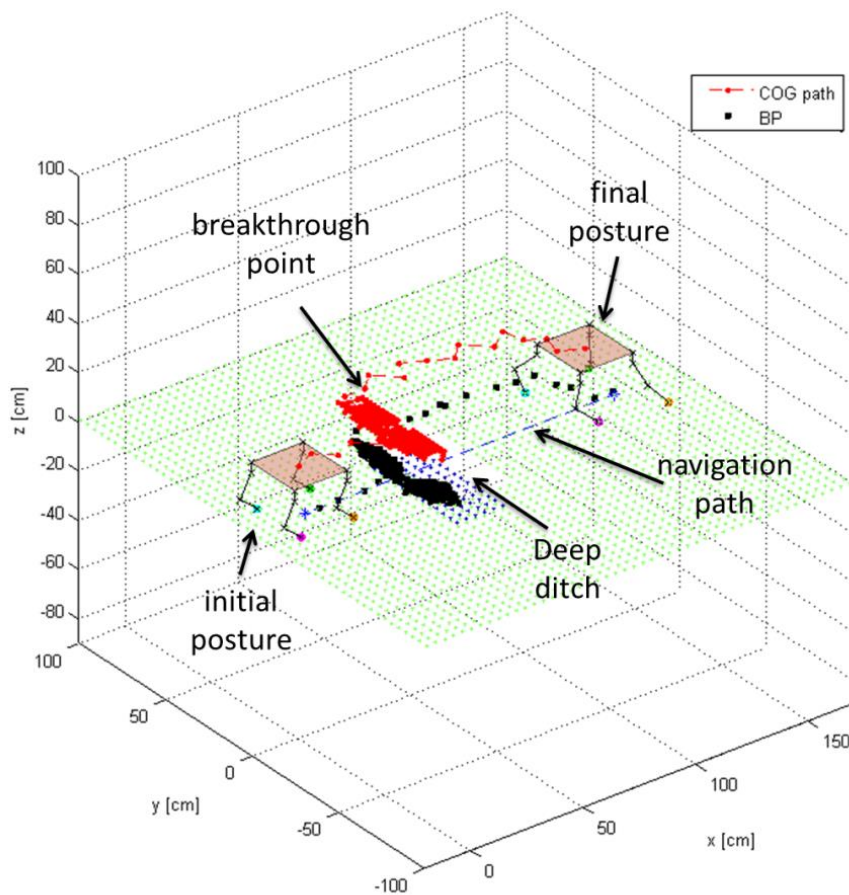
$$LPT = 0.07$$

$$Constraint_D_identity_threshold = 4.5[cm],$$

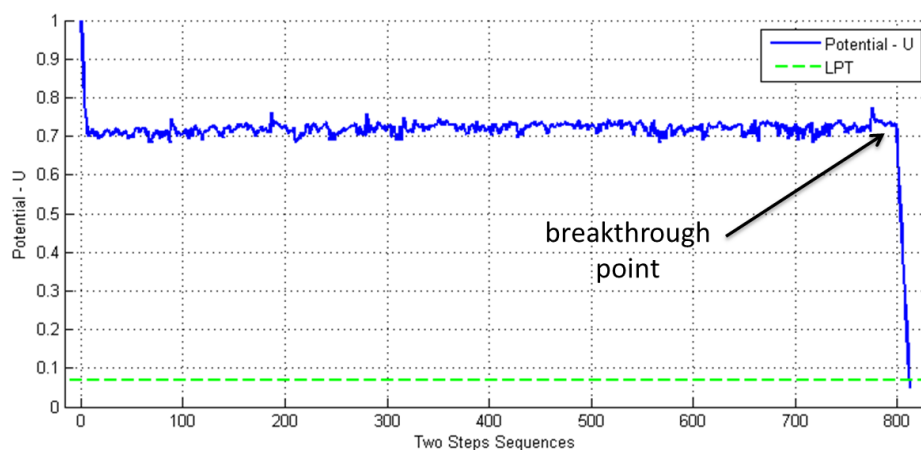
$$CKW_resolution = 2[cm]$$

$$w = [1, 0, 0]$$

תוצאות הסימולציה מוצגות באיורים הבאים :



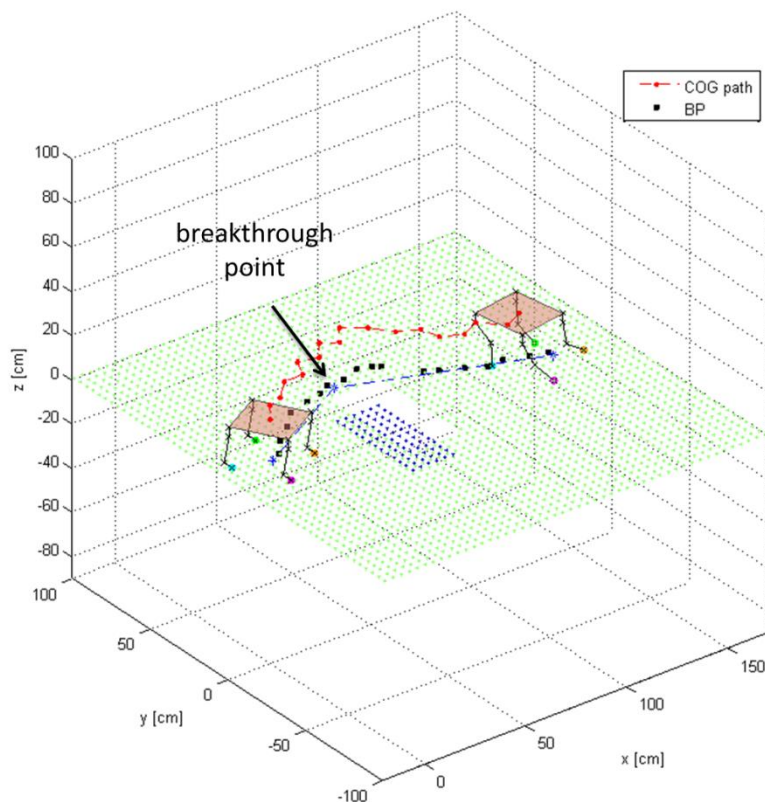
איור 26 - תוצאות סימולציה II – פלט האלגוריתם לפני הרצת השלב השני



איור 27 - תוצאות סימולציה II – שינוי הפוטנציאל לפני הרצת השלב השני

ניתן לראות באיור 26 שכעת מתקבל מסלול ניווט ארוך אשר כולל את שלילת כל הצעדים האפשריים לתנועה לפני מציאת נקודת המעבר, זהו שלב "מילוי הבור" של המינימום המקומי. באיור 27 ניתן לראות

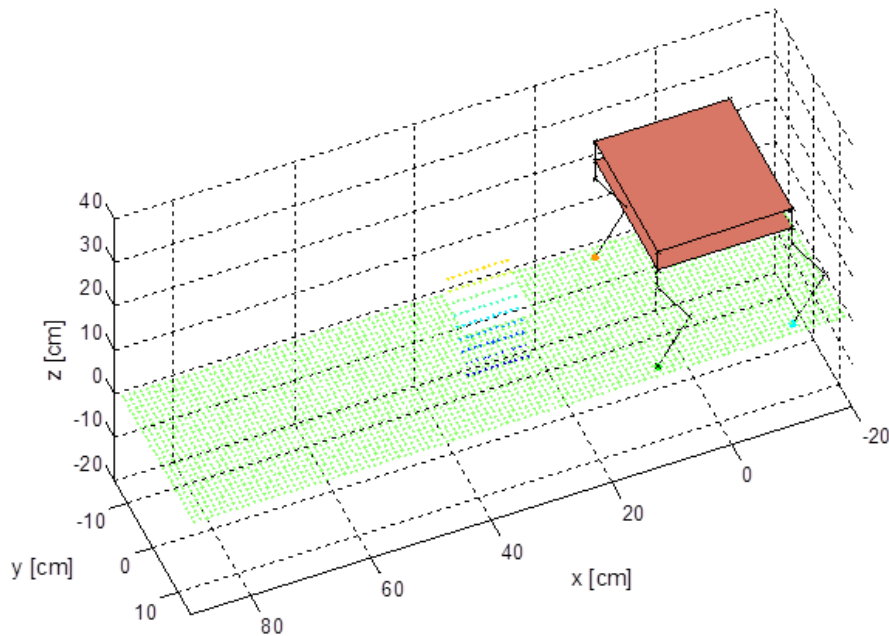
את שלב זה כחלק בו הפוטנציאל עולה במגמה איטית עד לנקודת ההיחלצות. כעת יש צורך בהפעלת השלב השני של האלגוריתם אשר ייצר מסלול הכולל נקודת ביניים בנקודת ההיחלצות מהמינימום המקומי (Breakthrough point). תוצאות השלב הראשון של האלגוריתם לאחר הרצתו עם המסלול המתוקן מופיעות באיור 28.



איור 28 - תוצאות סימולציה II – פלט האלגוריתם אחרי הרצת השלב השני

כעת מתקבל מסלול תקין אל נקודת היעד תוך ה"התחמקות" מבעיית המינימום המקומי. ניתן להפעיל את השלב השני פעמים נוספות עד לקבלת מסלול בו הפוטנציאל נמצא במגמת ירידה מונוטונית לאורך כל המסלול.

הסימולציה הבאה ממחישה את היתרון בשימוש בקריטריון היציבות FER בעת הליכה על קרקע מורכבת. לצורך פישוט נבחרה קרקע שטוחה בעלת אזור בודד בו השיפוע הוא 60 עם האופק.



איור 29 - סימולציה III – מודל הקרקע

האלגוריתם נדרש לתכנן הליכה על פני הקרקע לאורך 70 ס"מ תוך שמירה על גוף אופקי בגובה 22 ס"מ. בהרצה הראשונה האלגוריתם השתמש בקריטריון יציבות SSM בלבד ובהרצה השנייה הופעל קריטריון ה- FER . ערכי הפרמטרים אשר הוזנו לאלגוריתם בהרצת הסימולציה:

$$Grid_resolution = 3[cm]$$

$$MS = 2.5[cm]$$

$$FSF = 0.85$$

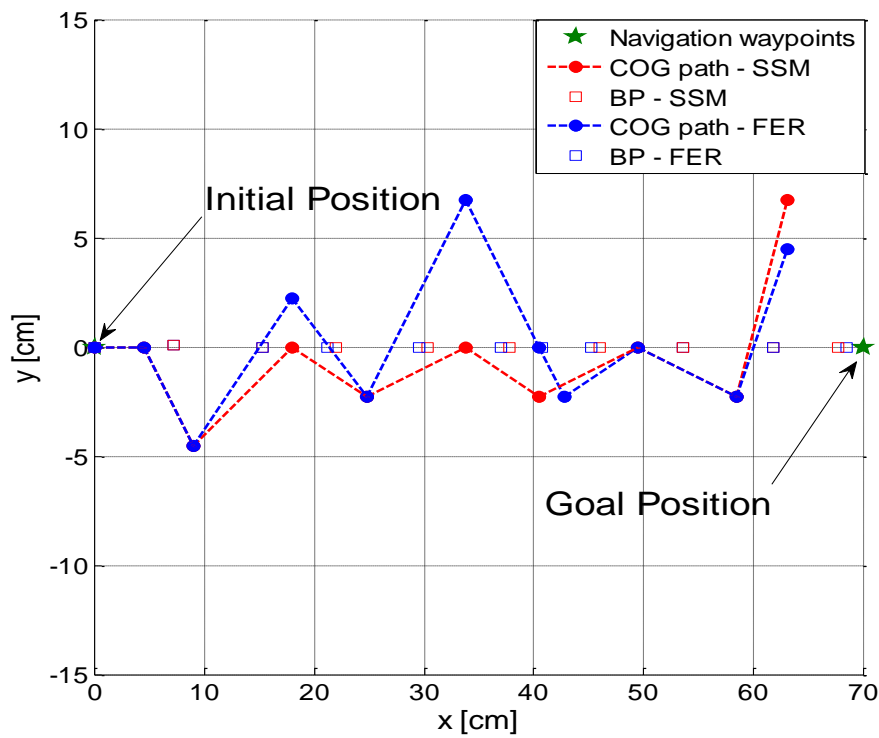
$$LPT = 0.05$$

$$Constraint_D_identity_threshold = 6.5[cm]$$

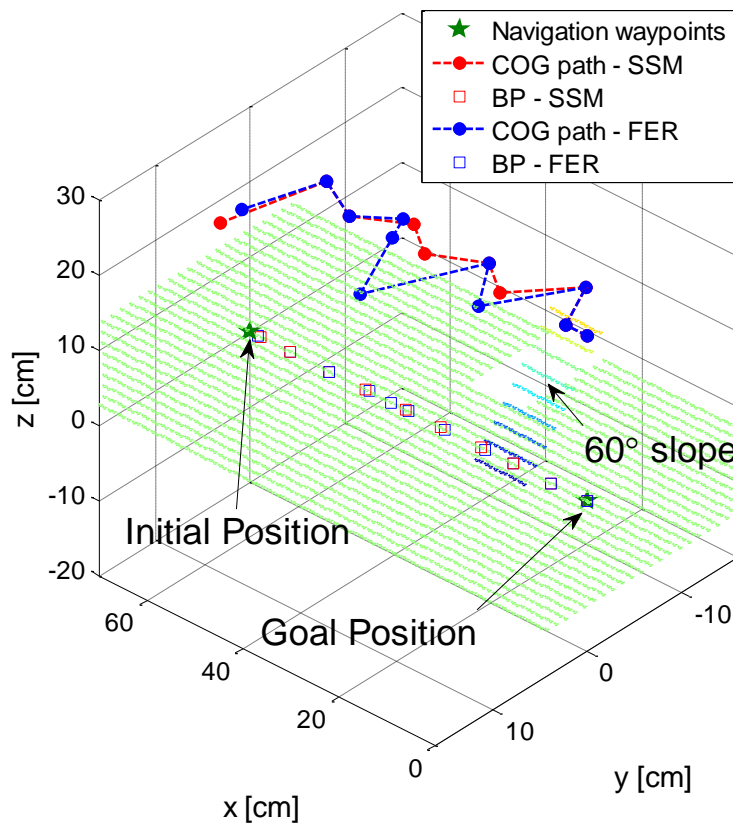
$$CKW_resolution = 2[cm]$$

$$w = [1, 0, 0]$$

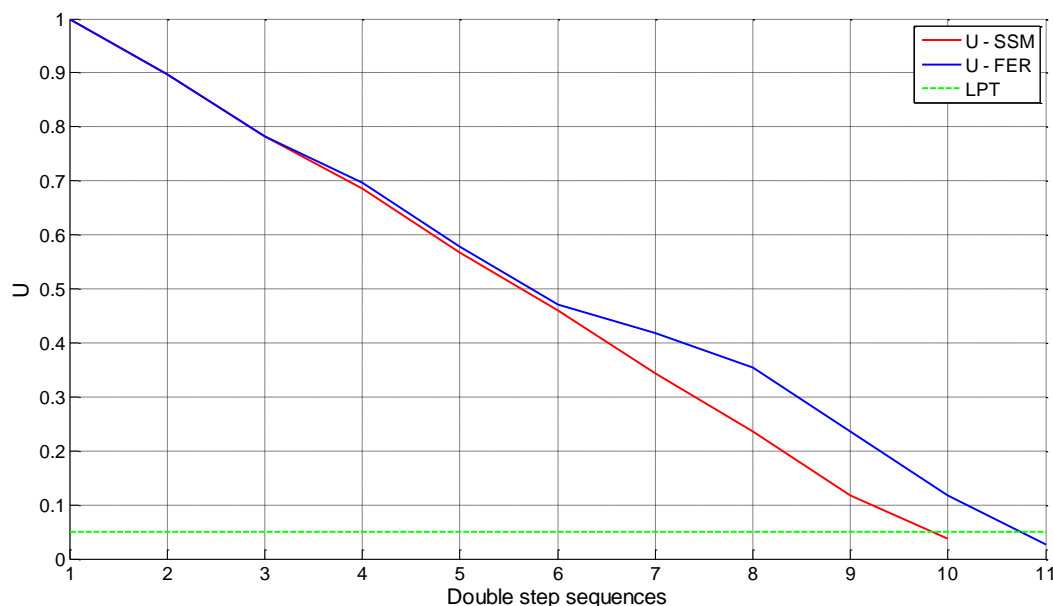
תוצאות הסימולציה מוצגות באיורים הבאים :



איור 30 - תוצאות סימולציה III – מסלול מרכז המסה וה-*BP* עבור הרצה עם קריטריוני יציבות שונים

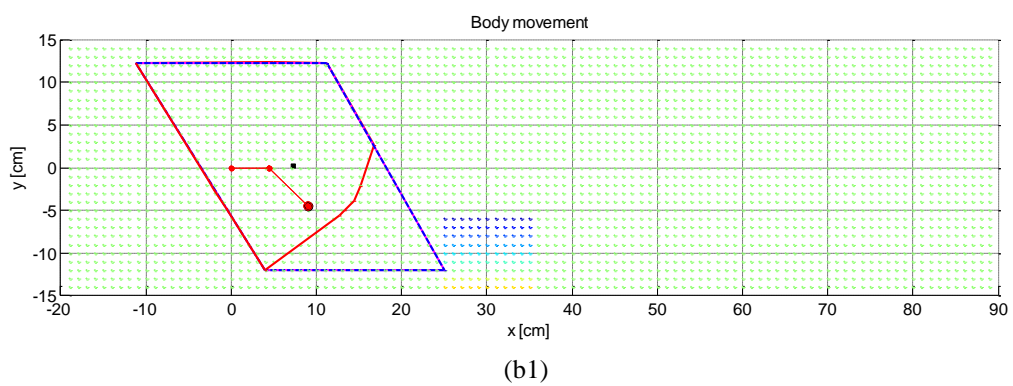
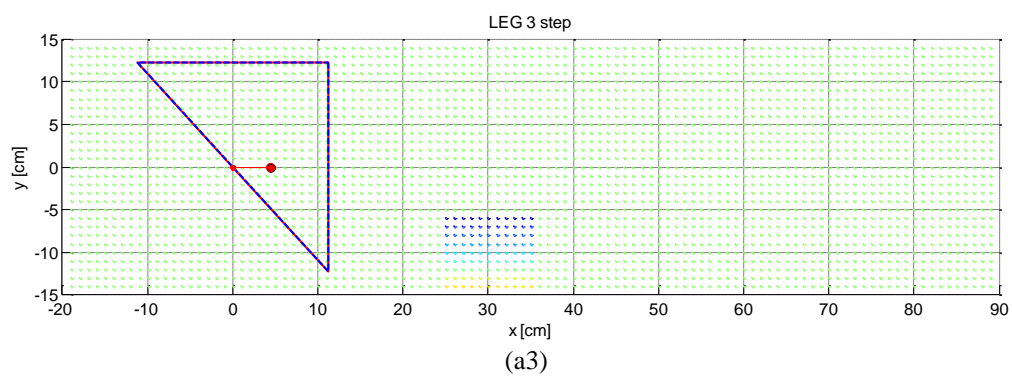
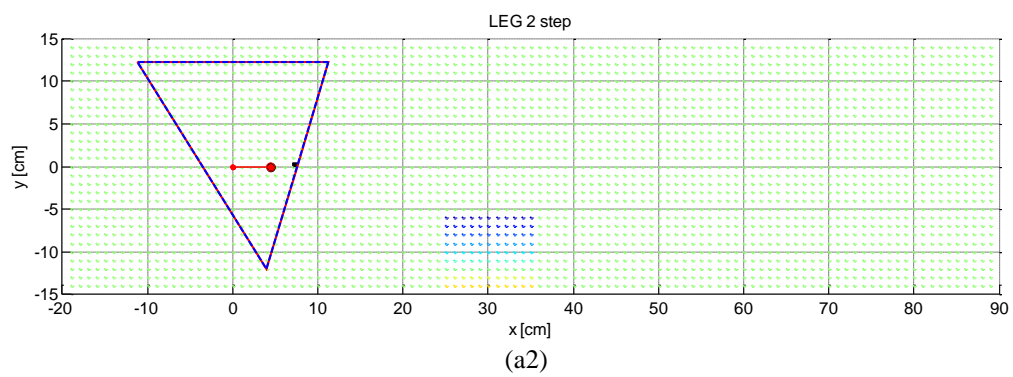
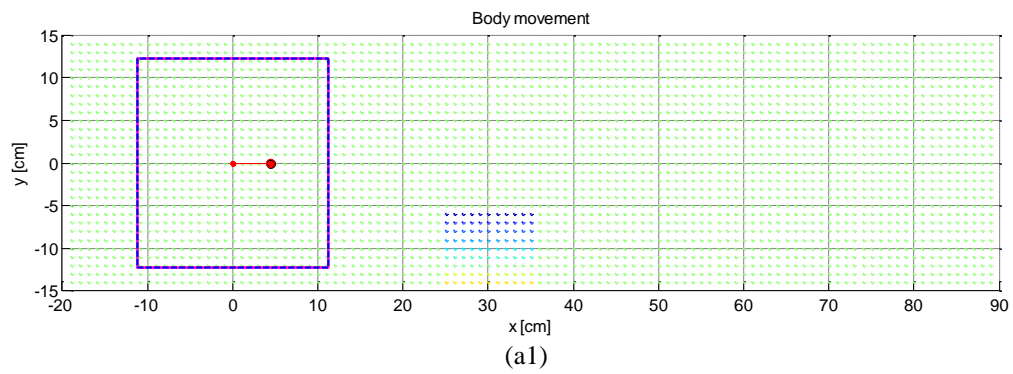


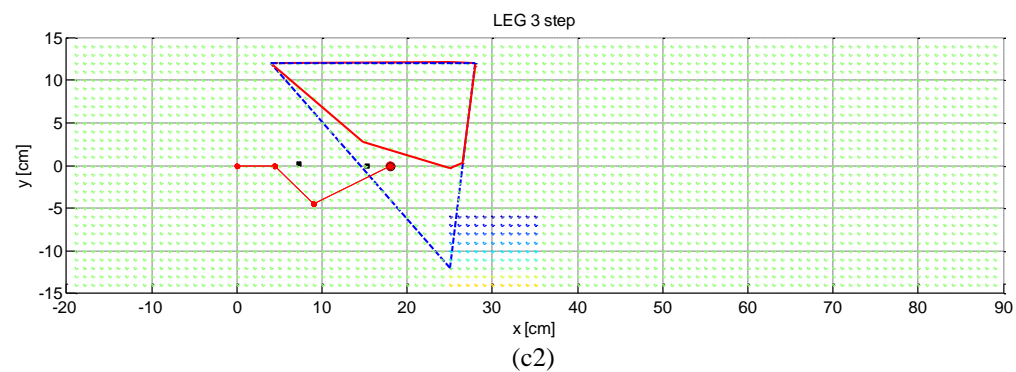
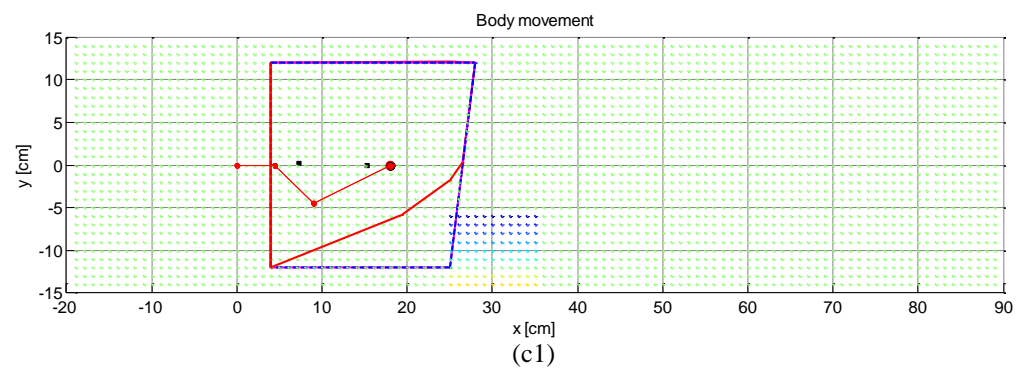
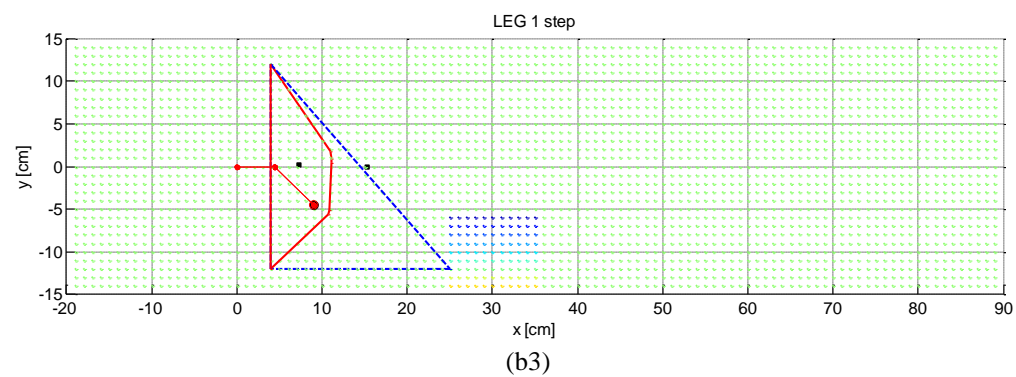
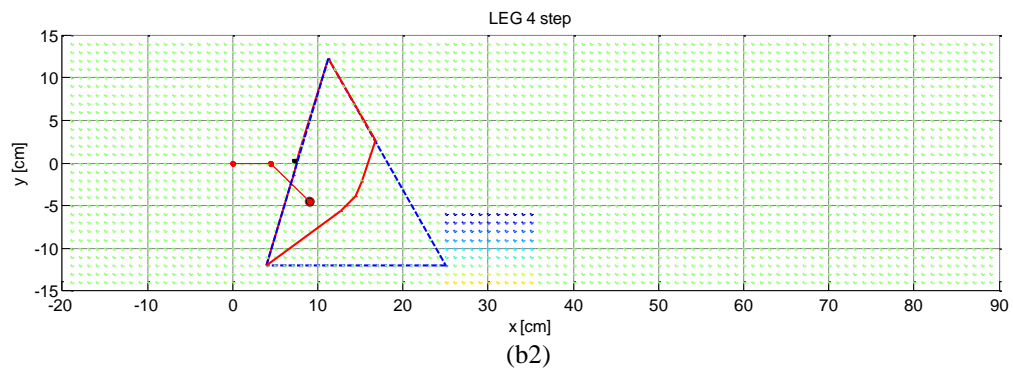
איור 31 - תוצאות סימולציה III – הצגת התוצאות, הקרקע, נקודת המוצא ונקודת היעד



איור 32 - תוצאות סימולציה III – שינוי הפוטנציאל עבור הרצה עם קריטריוני יציבות שונים

ניתן להבחין שבשימוש בקריטריון ה-*FER* מתקבל מסלול שונה למרכז המסה. באיור 30 ובאיור 31 ניתן לראות שבשני המקרים מתקבל מסלול המנווט את נקודת ה-*BP* לאורך המסלול תוך ירידה של הפוטנציאל (איור 32). ניתן גם לראות שבשימוש בקריטריון ה-*FER* התקבלו 11 צעדים כפולים לעומת 10 בשימוש ב-*SSM*. לצורך ניתח מעמיק יותר נציג את רצף התנועות אותן נדרש הרובוט לבצע כאשר בכל תנועה מוצג גם ה-*FER* וגם ה-*SP*. תנועה יכולה להיות תנועה של גוף הרובוט או ביצוע צעד. בעת הנעת הגוף ה-*SP* מורכב מ-4 נקודות המגע עם הקרקע בעוד שבביצוע צעד ה-*SP* מורכב מ-3 נקודות המגע של הגליים התומכות. איור 33 מציג את רצף התנועות כאשר מבוצע שימוש בקריטריון ה-*SSM* בלבד. באיור 33(a1) ניתן לראות שהתנועה מתחילה בהזזת גוף הרובוט (הקו האדום) לקראת ביצוע שני הצעדים בצעד הכפול. במהלך תנועה זו ארבעת רגלי הרובוט נמצאות על הקרקע ולכן ה-*SP* (המצולע הכחול) הוא בעל ארבע קודקודים. כעת, איור 33(a2), מתבצע צעד כאשר הרגל הדורכת היא רגל 3 (אחורית ימנית) ולכן ה-*SP* הופך למשולש. גוף הרובוט אינו נע בקטע זה. ב איור 33(a3) מוצג הצעד השני בצעד הכפול המתבצע ע"י רגל 2 (קדמית ימנית). בשלב זה רגל 3 כבר דורכת בנקודת המגע החדשה. לאחר שהצעד הכפול הראשון הושלם, מתחיל צעד כפול נוסף אשר מתחיל בהנעת גוף הרובוט, איור 33(b1). ניתן לשים לב שמכיוון שנקודת הדריכה של רגל 2 נמצאת על הקרקע המשופעת אין חפיפה מלאה בין ה-*SP* ל-*FER* (באדום). ניתן לראות שתנועת הגוף נשארת בתוך גבולות ה-*FER*. כעת, איור 33(b2), מתבצע הצעד הראשון באמצעות רגל 4 (אחורית שמאלית). ניתן לראות שגם לאחר המעבר ל-3 נקודות מגע עדיין קיימת עמידה בקריטריון היציבות. הצעד השני, איור 33(b3), מתבצע באמצעות רגל 1 (קדמית שמאלית). כעת מתבצע צעד כפול נוסף המתחיל בהנעת גוף הרובוט איור 33(c1). כעת הצעד הראשון בצעד הכפול מתבצע באמצעות רגל 3, איור 33(c2).

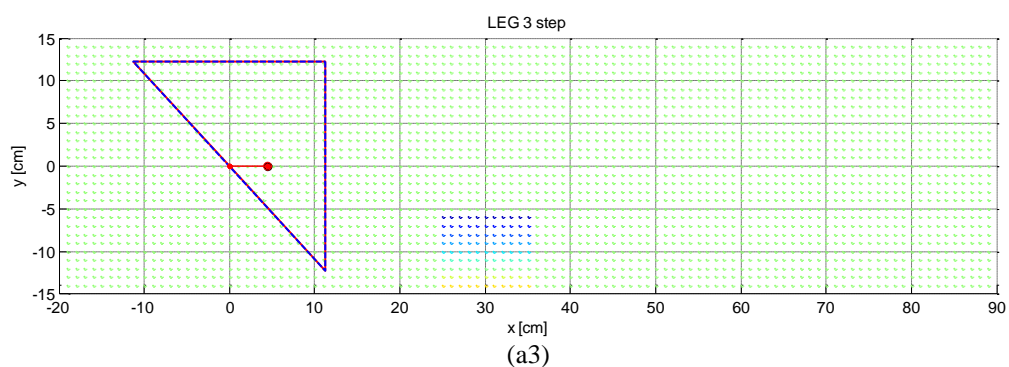
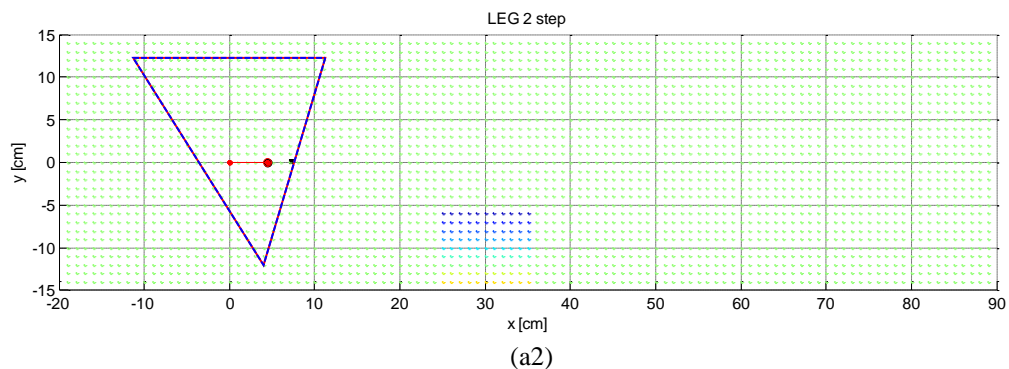
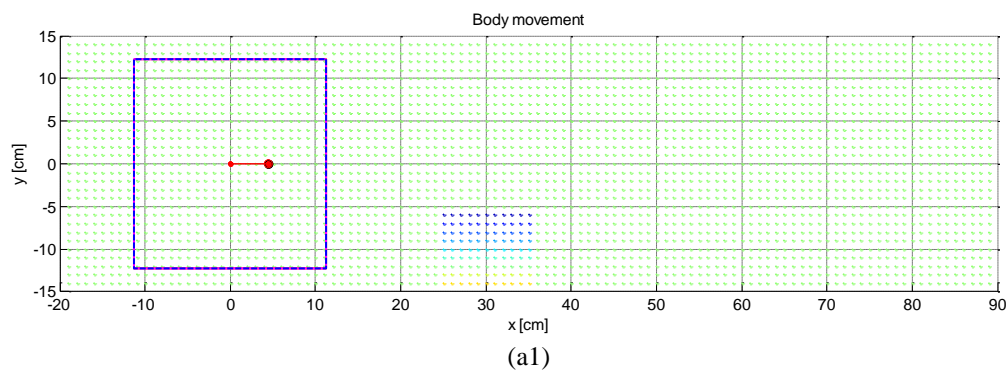


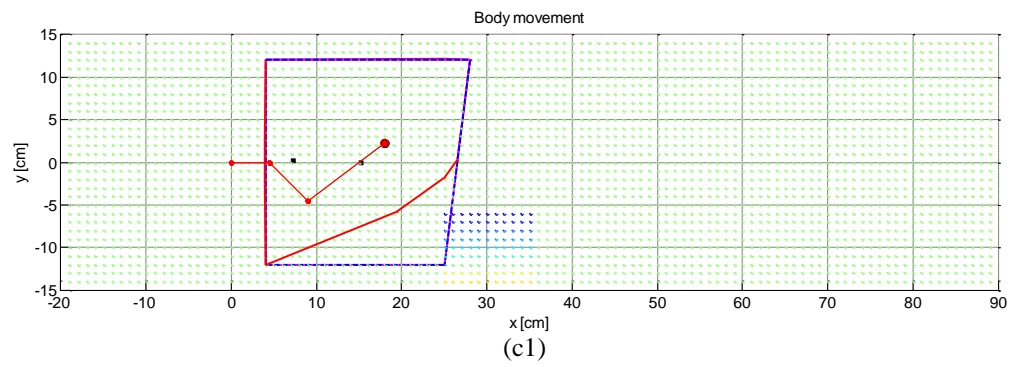
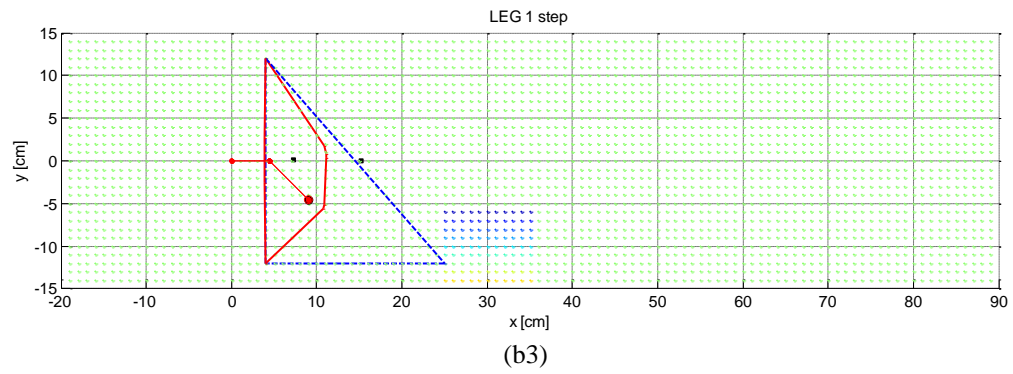
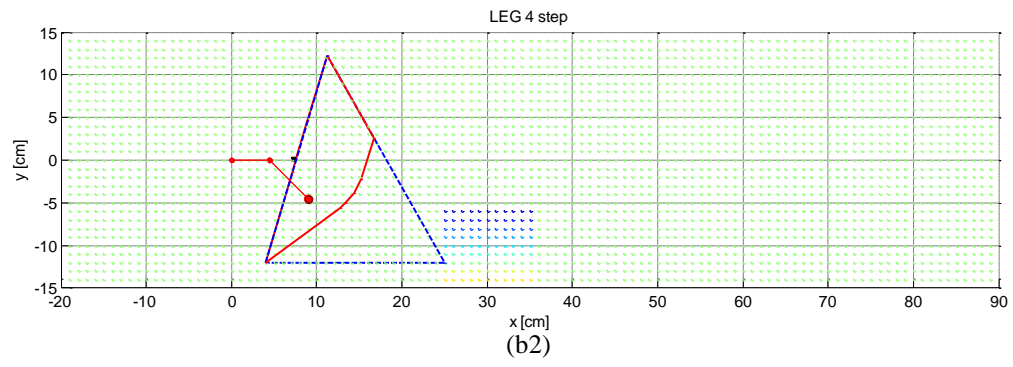
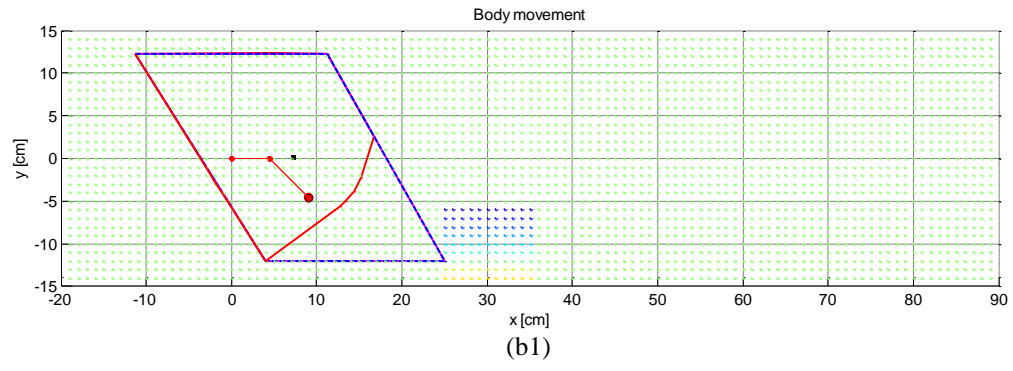


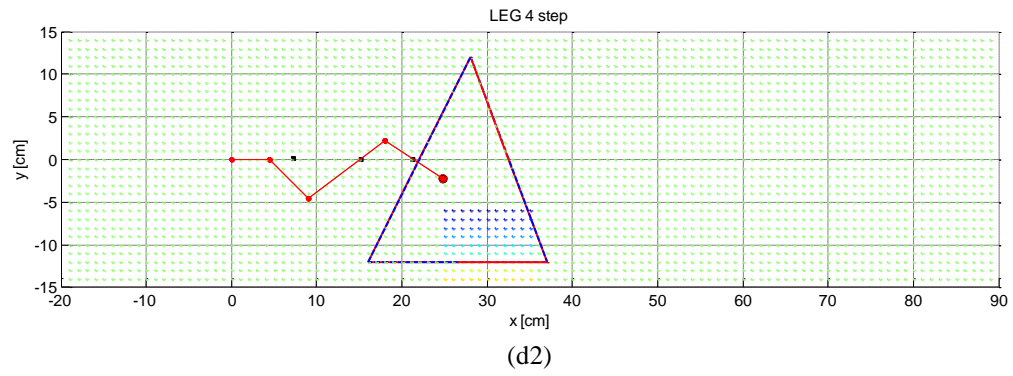
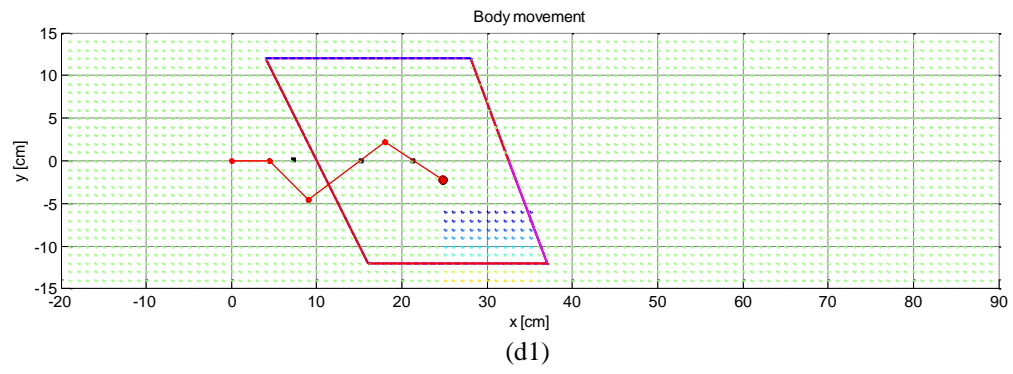
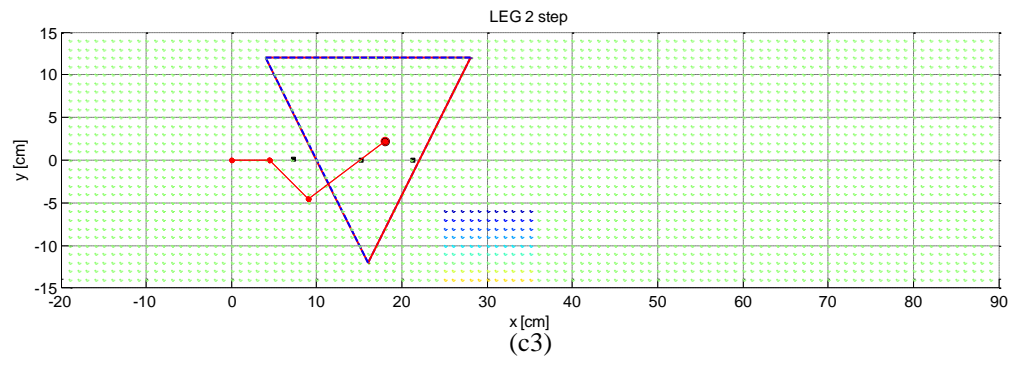
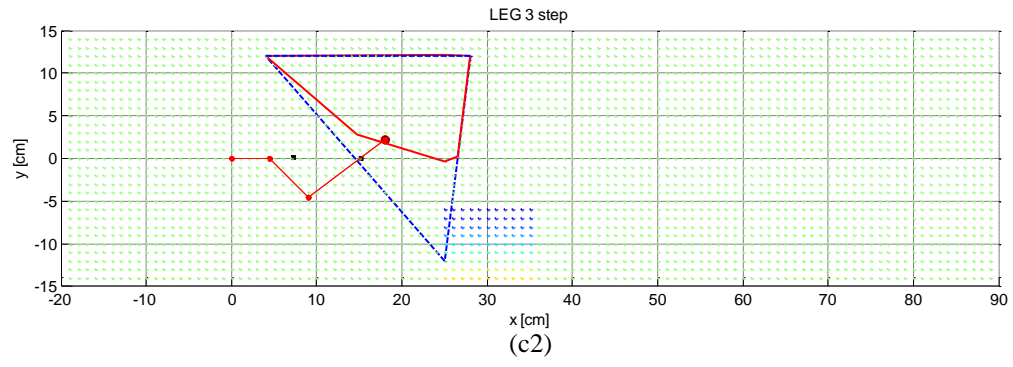
איור 33 - תנועות הרובוט בהרצה עם קריטריון SSM (מבט על)

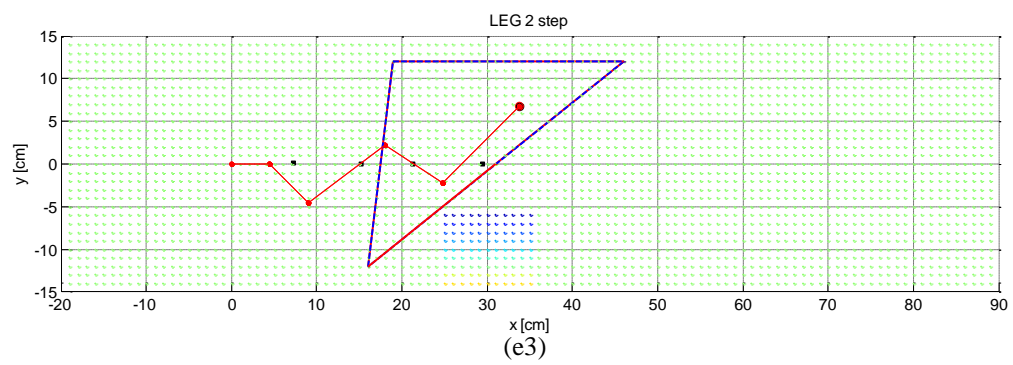
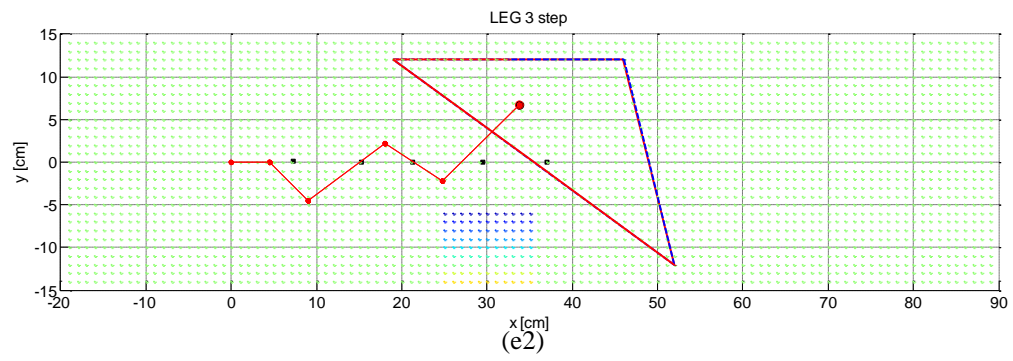
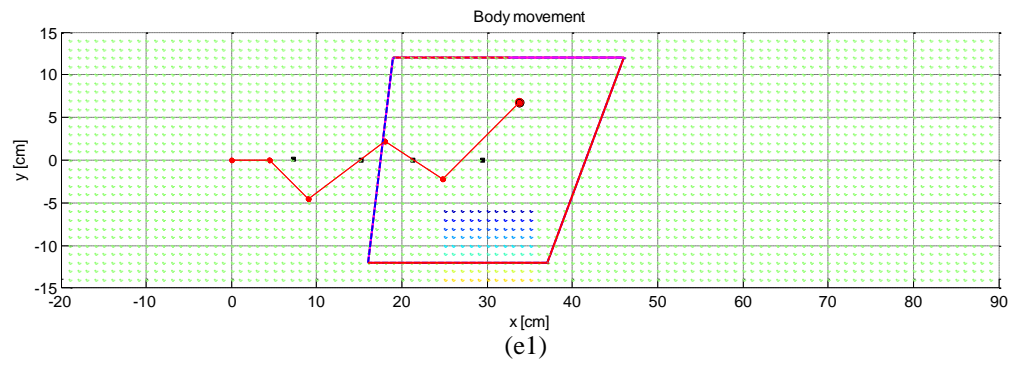
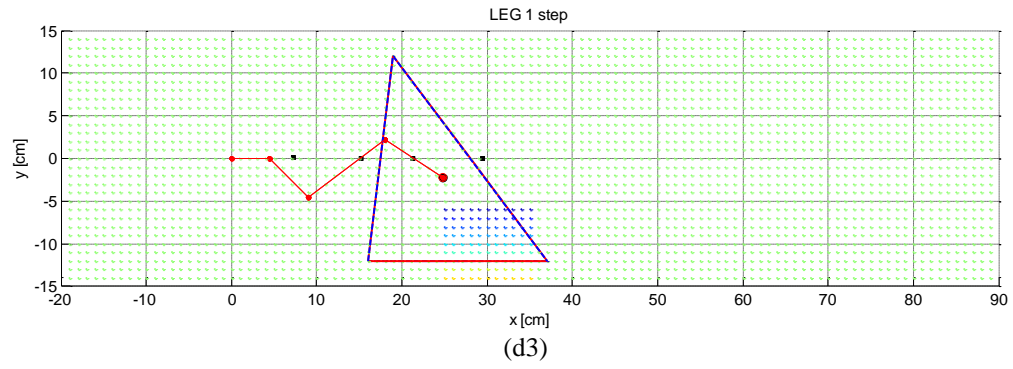
כפי שניתן לראות בתנועה האחרונה ברצף המוצג, מתקבל מצב בו אין עמידה בקריטריון ה-*FER*. כלומר, היטל מרכז המסה נמצא מחוץ לשטח ה-*FER*. פלט האלגוריתם המלא כולל גם את המשך תנועות הרובוט עד להגעה ליעד המכיל מקרים נוספים בהם קיימת חריגה משטח ה-*FER*. להמשך הפלט אין משמעות רלוונטית מכיוון שהרובוט ייפול עקב יציאה משיווי משקל לפני שיגיע לביצוע התנועות הבאות. בפרק 7.2.3 מוצג הניסוי אשר בוצע עבור רצף הפעולות הנ"ל עד לנפילת הרובוט.

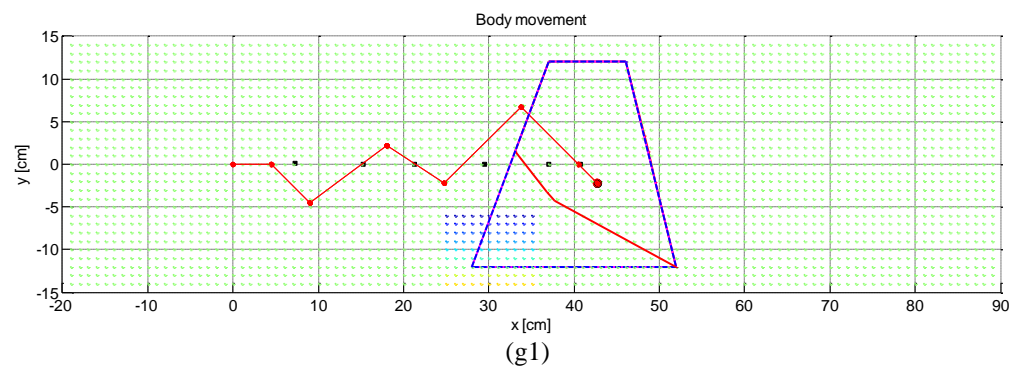
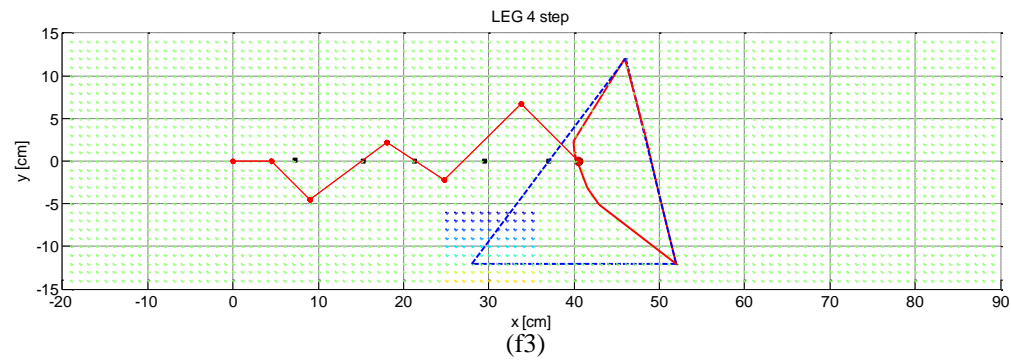
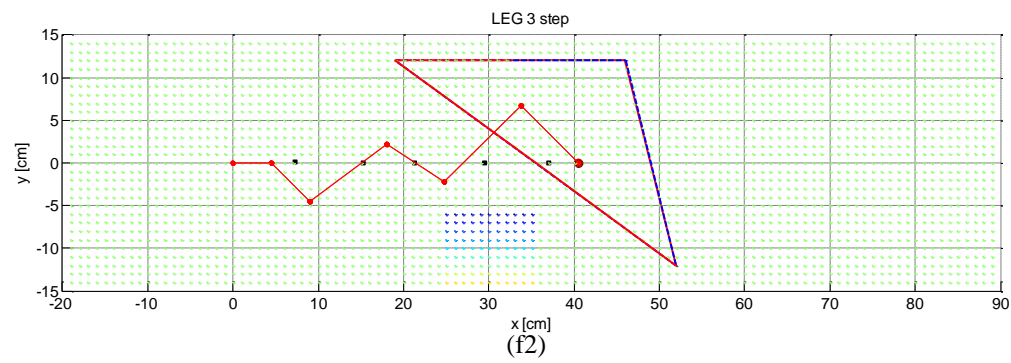
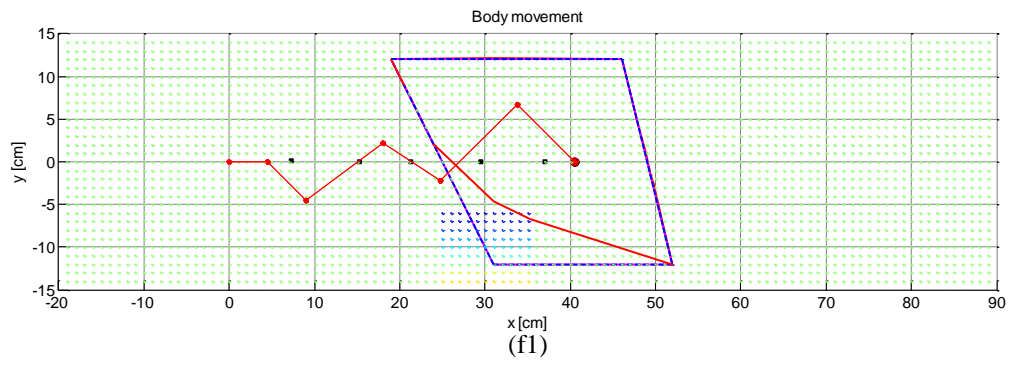
איור 34 מציג את סט התנועות שהאלגוריתם פלט כאשר התבצע שימוש בקריטריון ה-*FER*. האיור מתאר באופן דומה לאיור 33 את תנועות הרובוט כאשר התנועה מתחילה בהנעת גוף הרובוט, תנועת רגל ראשונה בצעד הכפול ולאחר מכן תנועת הרגל השנייה. רצף זה חוזר עד להגעה לנקודת היעד.

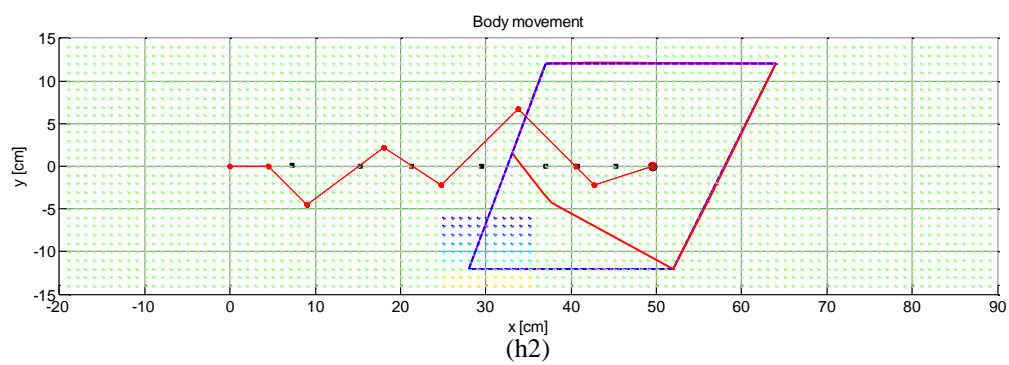
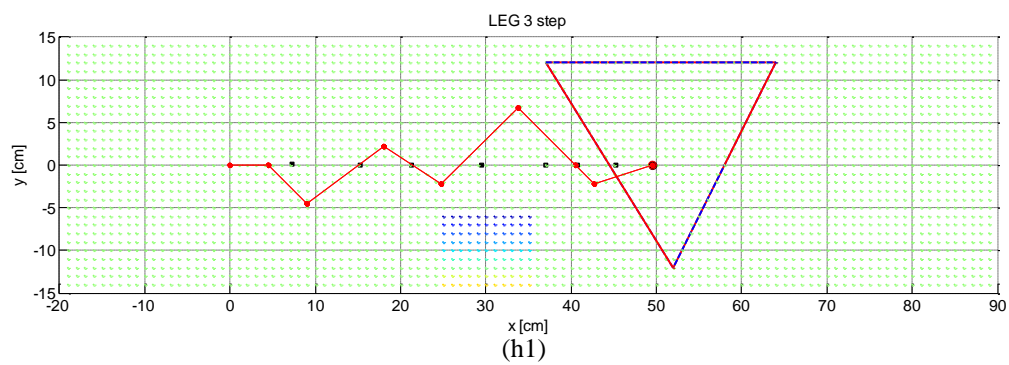
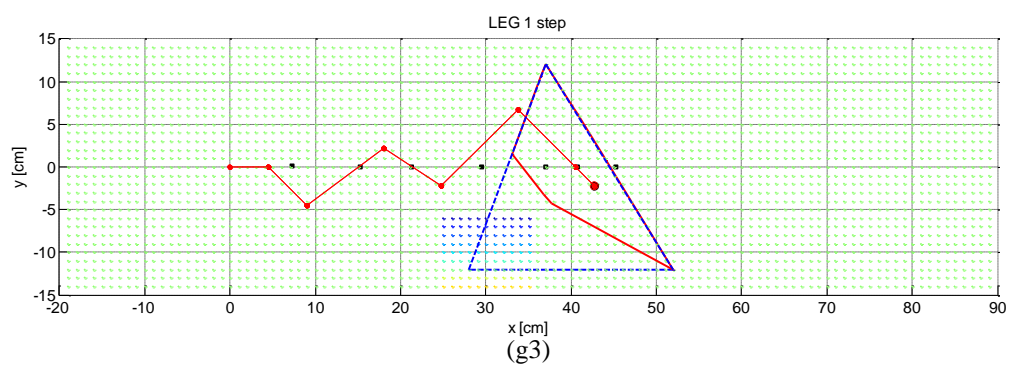
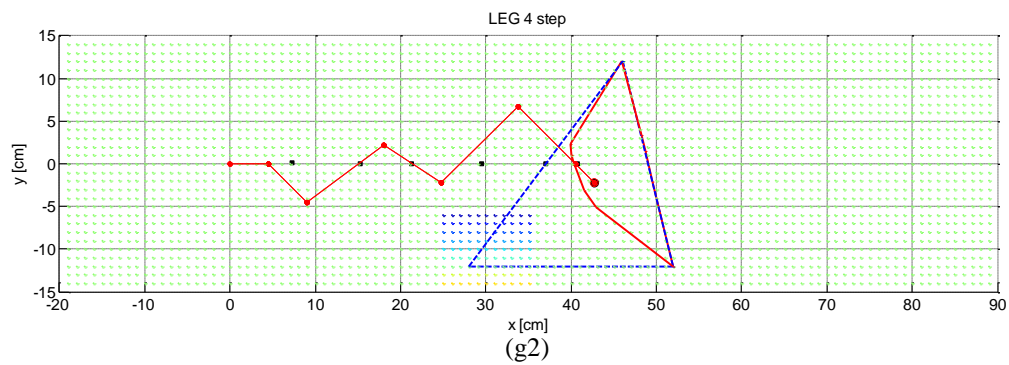


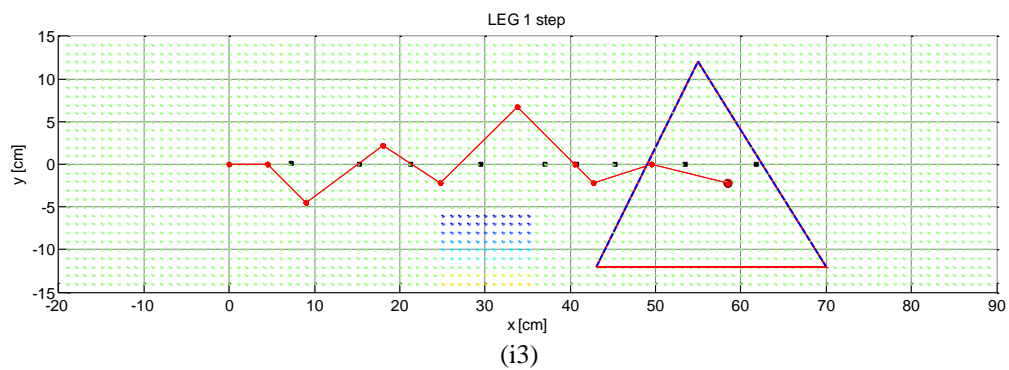
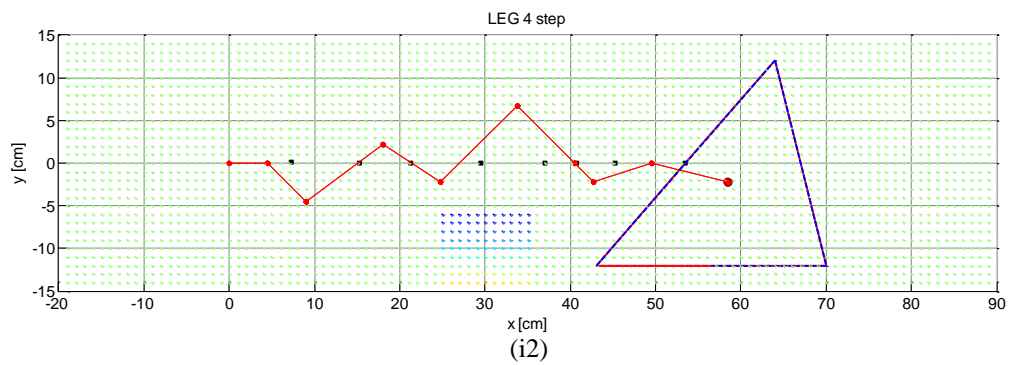
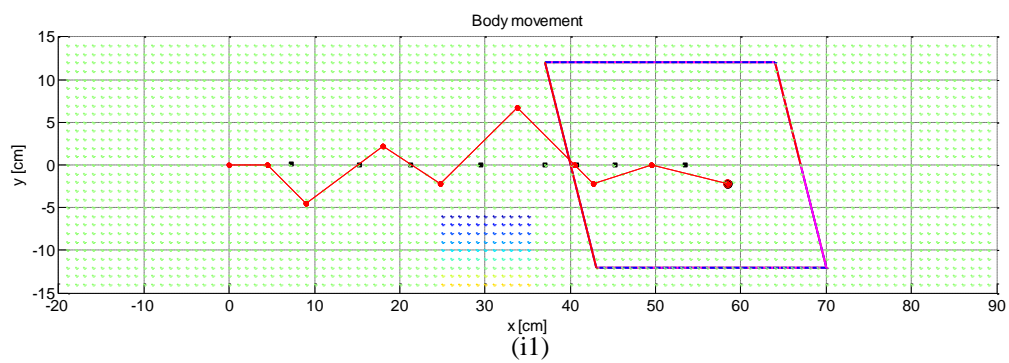
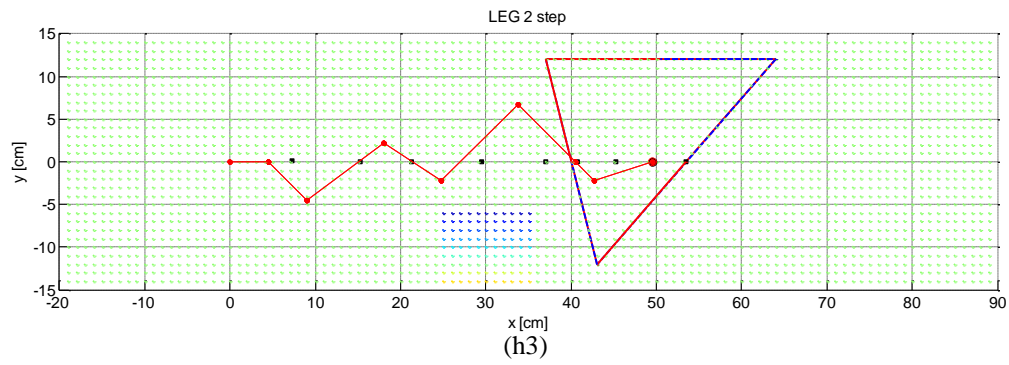


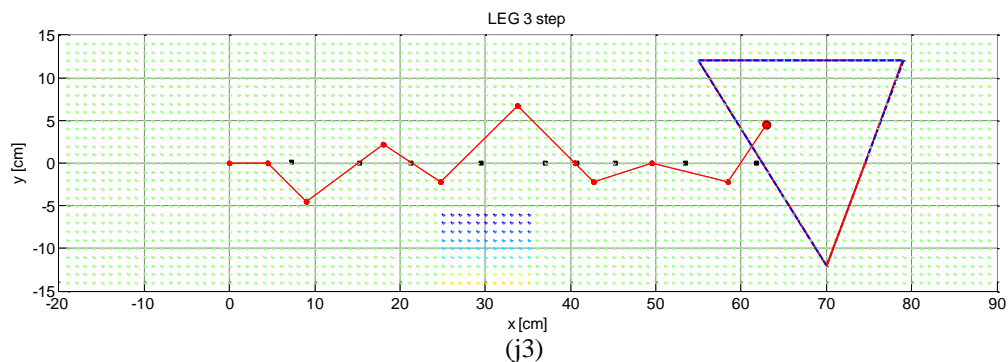
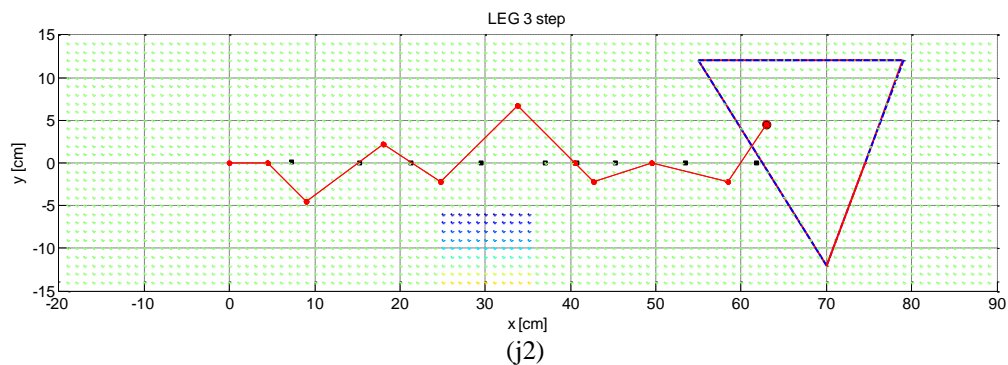
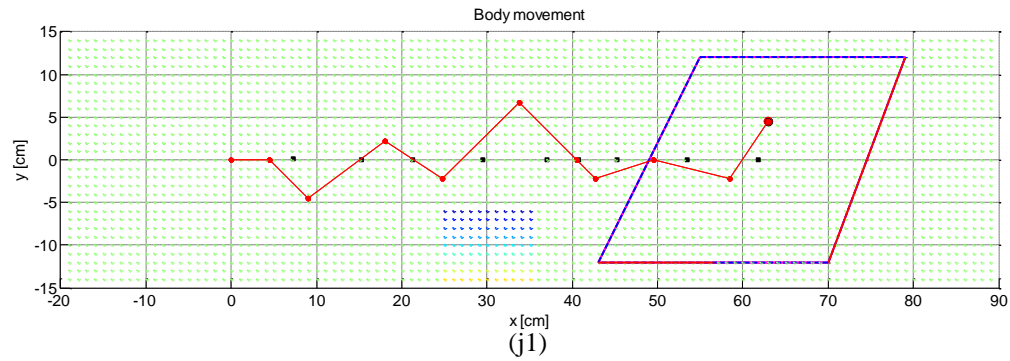












איור 34 - תנועות הרובוט בהרצה עם קריטריון FER (מבט על)

כעת רצף התנועות כולו מתבצע תוך שמירה על קריטריון היציבות FER . יש לציין שגם בחלקי התנועה בהן מרכז המסה קרוב לשפת ה- FER , כמו למשל באיור 34(c2) ובאיור 34(f3), אין חשש לפגיעה בשיווי המשקל עקב השימוש במקדם הבטיחות FSF . בפרק 7.2.3 מוצג הניסוי שנעשה עבור רצף תנועות זה.

כפי שהוסבר בפרק 5, האלגוריתם "מתייחס" באופן זהה לנקודות קרקע משופעת ולנקודות קרקע מישורית. לכן, המסלול הנבחר הוא זה הקצר ביותר והעובר דרך נקודות קרקע משופעת. זאת למרות שקיים מסלול בו מתבצעות דריכות רק על קרקע מישורית.

משך זמן החישוב הממוצע עבור צעד כפול אחד בשימוש בקריטריון ה- SSM בלבד הוא 2.9 שניות בעוד ששימוש בקריטריון ה- FER העלה את זמן החישוב ל-4.4 שניות. חשוב לציין שבשטח מורכב יותר הפרש זמני החישוב יהיה משמעותי יותר עקב ריבוי חישובים של קריטריון ה- FER . כמו כן, סימולציה זו

ממחישה בבירור את היתרון בשימוש בקריטריון ה- FER אשר הביא לכך שפלט האלגוריתם מבטיח שמירה על שיווי משקל בכל אחד משלבי ההליכה.

6.2.4 סימולציה IV

סימולציה זו מדגימה את יכולת תכנון התנועה על פני קרקע מורכבת. ערכי הפרמטרים אשר הוזנו לאלגוריתם בהרצת הסימולציה:

$$Grid_resolution = 4[cm]$$

$$MS = 2.5[cm]$$

$$FSF = 0.85$$

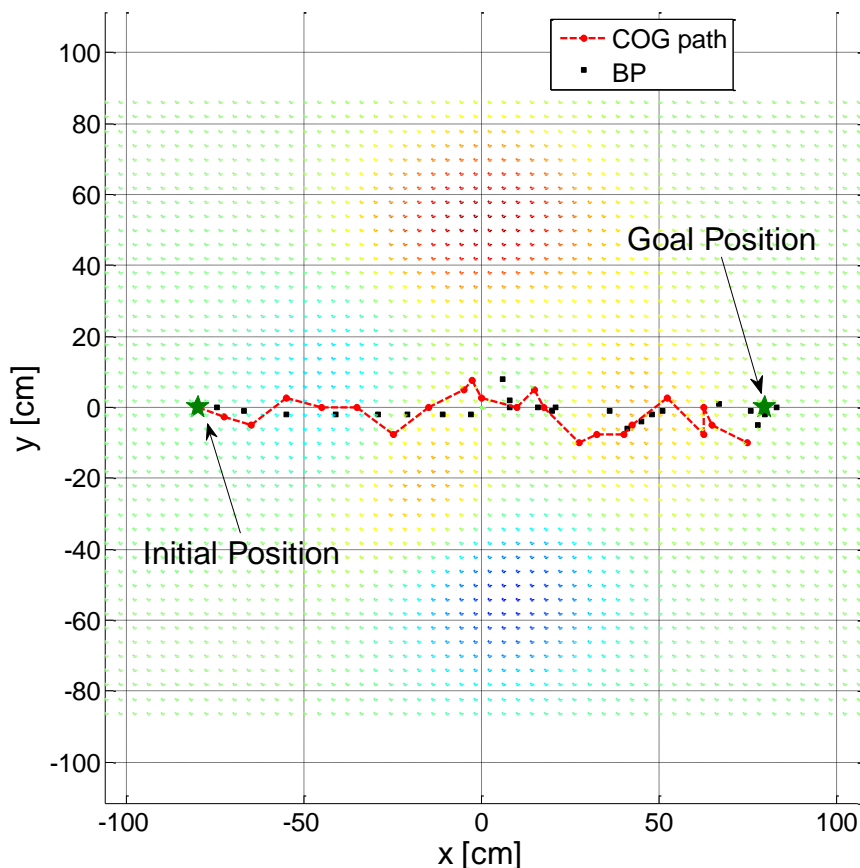
$$LPT = 0.03$$

$$Constraint_D_identity_threshold = 3.5[cm]$$

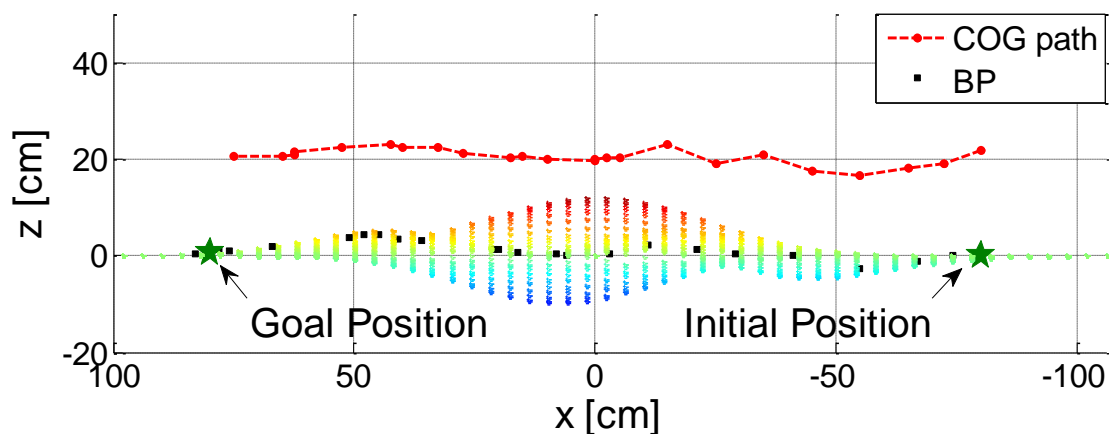
$$CKW_resolution = 2[cm]$$

$$w = [1, 0, 0]$$

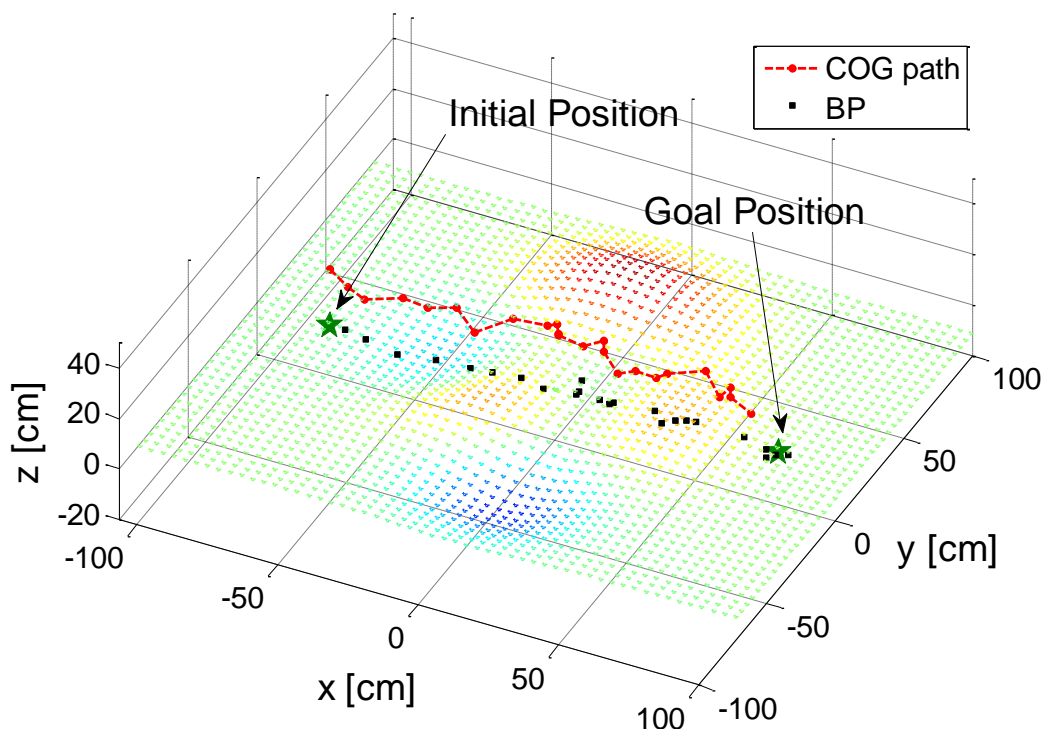
נקודת היעד הוגדרה במרחק 80 ס"מ מנקודת ההתחלה והאלגוריתם נדרש לתכן את התנועה שתעביר את הרובוט על פני השטח המורכב תוך שמירה על שיווי המשקל. איור 35, איור 36 ואיור 37 מציגים את המסלול שהתקבל ממבט על, מבט צד ומבט תלת-ממדי בהתאמה.



איור 35 - תוצאות סימולציה IV – מסלול מרכז המסה – מבט על

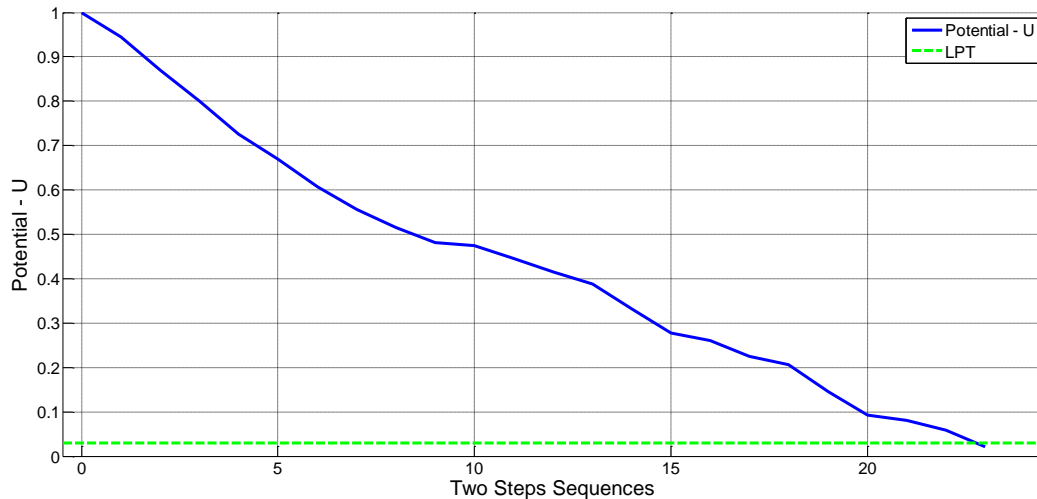


איור 36 - תוצאות סימולציה IV – מסלול מרכז המסה – מבט צד



איור 37 - תוצאות סימולציה IV – מסלול מרכז המסה – מבט תלת-ממדי

המסלול המוצג באדום הינו המסלול אותו עובר מרכז המסה של הרובוט עד להגעה אל נקודת היעד. ניתן לראות שבנוסף לתנועות מצד לצד, מרכז המסה גם נע מעלה ומטה כדי לאפשר לרובוט טיפוס או ירידה לאורך הקרקע המשופעת. איור 38 מציג את השינוי בפוטנציאל לאורך ההתקדמות.



איור 38 - תוצאות סימולציה IV – שינוי הפוטנציאל לאורך התקדמות הרובוט

ניתן לראות שעקב מורכבות הקרקע הירידה בפוטנציאל אינה מונוטונית ו"חלקה" אך בסופו של דבר הפוטנציאל יורד מתחת לערך ה- LPT כאשר הרובוט מגיע ליעדו. סימולציה זו הינה דוגמא לפלט האלגוריתם כאשר נדרשת התמודדות עם קרקע מורכבת שאינה מישורית.

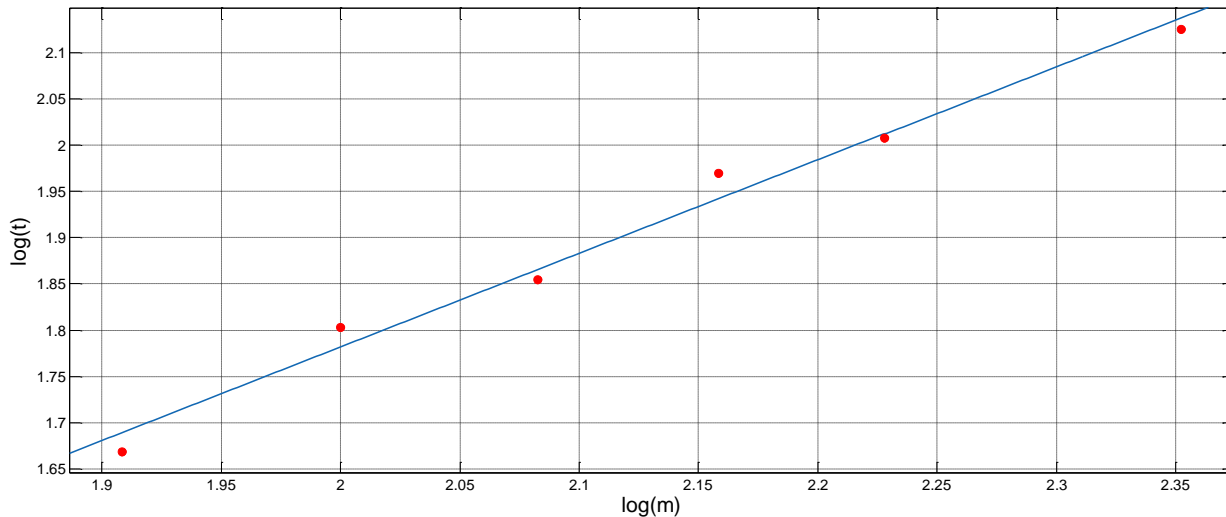
6.2.5 סימולציה V

סימולציה זו נועדה לבחון את השפעת פרמטרים הניתנים לשליטה על זמן הריצה של האלגוריתם. הפרמטרים הנחקרים הם צפיפות מרחב ה- CKW וצפיפות מרחב ה- LWA . כפי שמפורד בפרק 5, הפרמטר m מציין את מספר הנקודות במרחב ה- CKW והפרמטר n מציין את מספר הנקודות במרחב ה- LWA . בסימולציה הורץ האלגוריתם כאשר הרובוט נדרש להתקדם 20 ס"מ קדימה על קרקע מישורית. בהרצות הראשונות אילוף C (אילוף המבטיח שיהיה אפשר לבצע את הצעד הבא באמצעות כל אחת מהרגליים) בוטל ומכאן סיבוכיות האלגוריתם הינה $O(mn^2k^4)$. בשלב הראשון הורץ האלגוריתם 6 פעמים כאשר בכל הרצה שונה הפרמטר m בעוד ש- n ו- k (מספר הנקודות בגריד הקרקע) נשארו קבועים. לצורך הערכת השפעת השינוי נעזר בקשר הבא,

$$t = C(mn^2k^4) = C'm \quad (27)$$

$$\log(t) = \log(C') + \log(m) = C'' + \log(m)$$

כאשר t הוא זמן הריצה הכולל של האלגוריתם. כלומר, מצופה שהיחס בין $\log(t)$ ל- $\log(m)$ יקיים קשר לנארי של משוואת ישר בעל שיפוע בגודל 1. התוצאות מוצגות באיור 39.



איור 39 - השפעת צפיפות מרחב ה-CKW על זמן הריצה כאשר אילוף C מבוטל

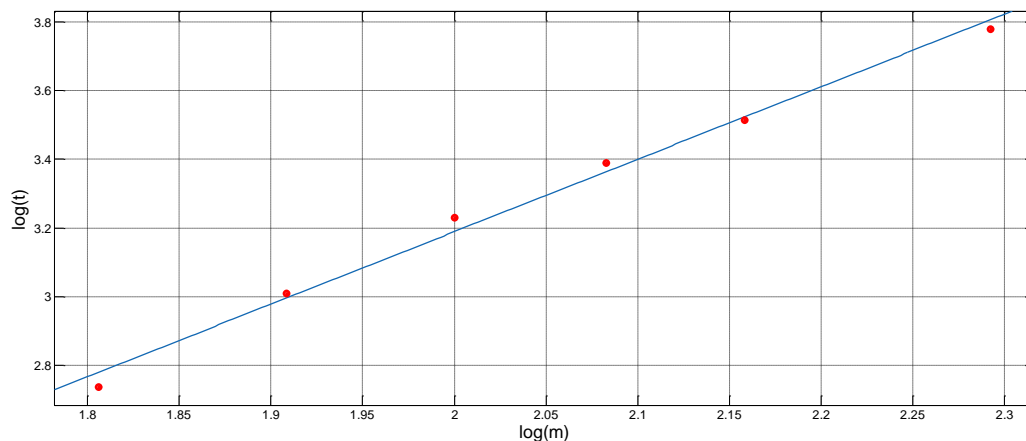
מקירוב ריבועים פחותים למשוואת ישר (מסומן בקו כחול) מתקבל שיפוע בגודל 1.0123 (טיב הקירוב $R^2 = 0.9819$). כלומר, השפעת הפרמטר m הינה אכן מסדר ראשון.

כעת הופעל אילוף C אשר מעלה את הסיבוכיות ל- $O(m^2 n^2 k^4)$ ואת השיפוע המצופה ל-2.

$$t = C(m^2 n^2 k^4) = C' m^2 \quad (28)$$

$$\log(t) = \log(C') + 2\log(m) = C'' + 2\log(m)$$

התוצאות עבור 6 הרצות נוספות מוצגות באיור 40.



איור 40 - השפעת צפיפות מרחב ה-CKW על זמן הריצה כאשר אילוף C מופעל

מהקירוב למשוואת ישר (מסומן בקו כחול) מתקבל שיפוע בגודל 2.1091 (טיב הקירוב $R^2 = 0.9923$). זהו ערך קרוב מאוד לערך המצופה - 2.

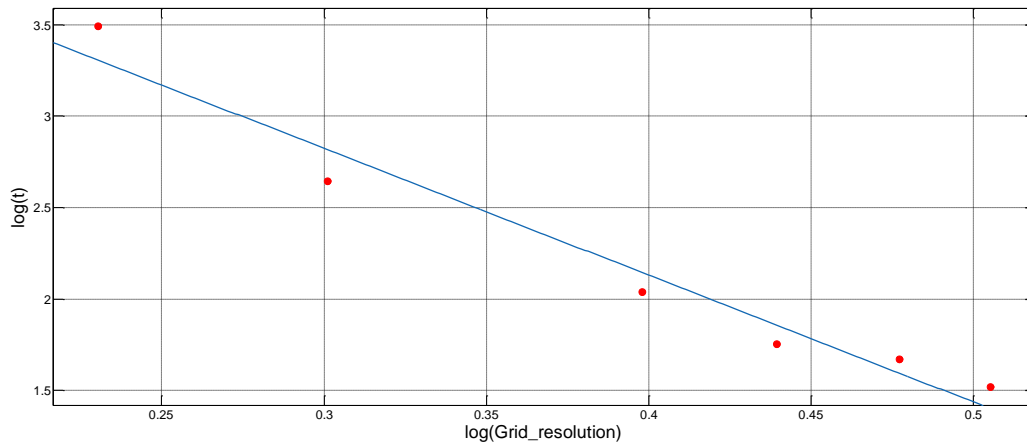
בהרצות הבאות הפרמטר ששונה הוא רזולוציית גריד הקרקע. כפי שמפורט בפרק 5, לפרמטר זה

השפעה הן על הפרמטר k והן על הפרמטר n . נשתמש שוב בקשר הבא:

$$t = C \left(m / \text{Grid_resolution}^{12} \right) = C' \cdot \text{Grid_resolution}^{-12} \quad (29)$$

$$\log(t) = \log(C') - 12 \log(\text{Grid_resolution}) = C'' - 12 \log(\text{Grid_resolution})$$

מכאן שהשיפוע המצופה הוא שיפוע שלילי בגודל 12. השיפוע הוא שלילי מכיוון שככל שהמרווח בין נקודות קטן כך קיימות יותר נקודות באותו השטח והפרמטרים k ו-n גדלים. איור 41 מציג את התוצאות שהתקבלו עבור 6 רזולוציות גריד שונות.



איור 41 - השפעת רזולוציית הגריד על זמן הריצה

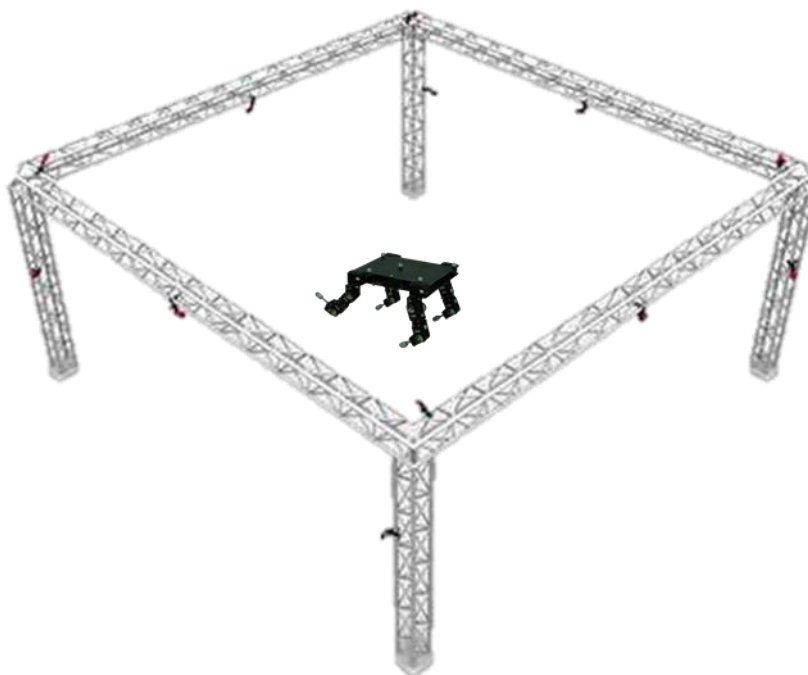
השיפוע שהתקבל מקירוב לקו ישר (מסומן בכחול) באמצעות ריבועים פחותים הוא -6.9357 (טיב הקירוב

$R^2 = 0.9627$). זהו ערך נמוך מהערך המצופה. כלומר, הסיבוכיות בהרצות שהתבצעו נמוכה יותר מסיבוכיות החסם העליון. הסיבה לכך היא שערך החסם העליון חושב עבור המקרה הגרוע ביותר. במקרה זה לא קיים מסלול והחיפוש בגרף מתבצע בגרף כולו. לעומת זאת, בכל המקרים המתוארים בסימולציות הני"ל קיים מסלול אשר נמצא בזמן הקצר משמעותית מהזמן הלוקח לסרוק את הגרף כולו.

7 ניסויים

7.1 מערכת המדידה החיצונית

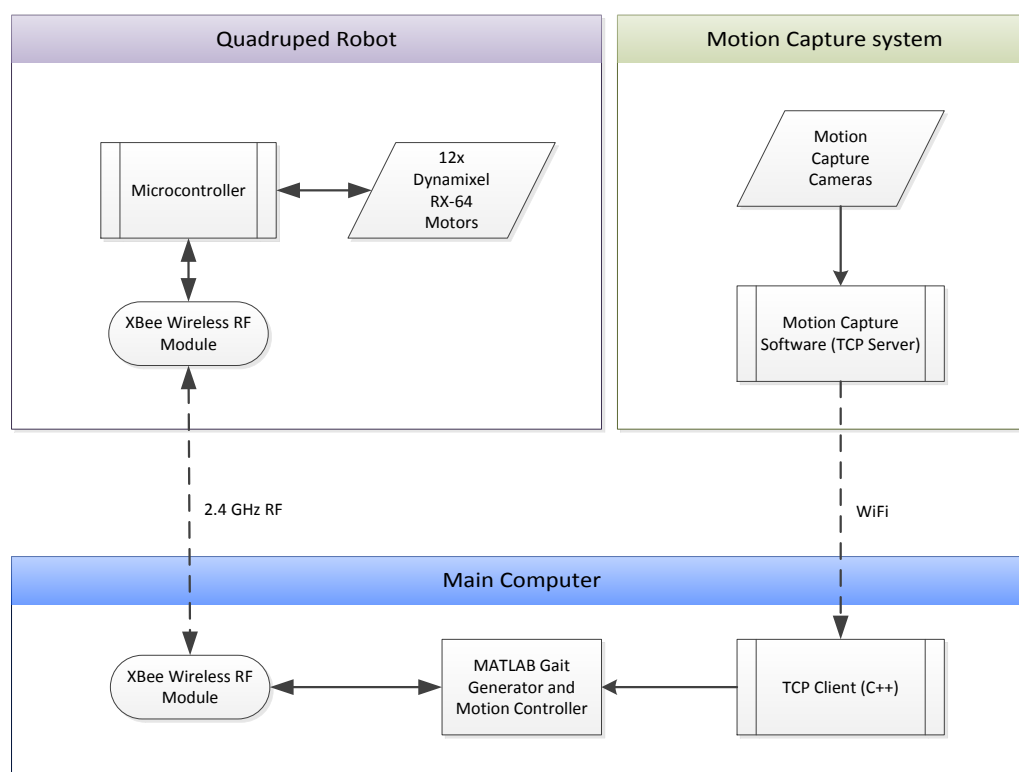
על מנת לבצע הערכה והפקה של מסקנות כלשהן מתוצאות הניסוי יש להשתמש במערכת מדידה אשר באמצעותה ניתן לכמת את תוצאות הניסוי. מערכת המדידה משמשת בחיפוש מיקום המשמש לקבלת המיקום הנוכחי של הרובוט בזמן אמת. מערכת המדידה שנבחרה היא מערכת מצלמות ללכידת תנועה (Motion capture) מסוג OptiTrack של חברת NaturalPoint. המערכת מורכבת ממצלמות המקרינות קרינת אינפרא-אדום וקולטות את ההחזרים מסמנים הממוקמים על האובייקט שנמצא תחת עקיבה. איור 42 מציג אילוסטרציה למערכת המדידה. המידע מכל המצלמות מתרכז במחשב אשר מבצע עיבוד על המידע הכולל ומחשב את מיקומו המדויק של כל סמן בדיוק גבוה מאוד (פחות ממילימטר). ככל שיותר מצלמות מזהות את אותו הסמן כך גדל הדיוק בחישוב מיקומו.



איור 42 - אילוסטרציה למיקום מצלמות סביב זירת ניסוי

הניסויים נמדדו במערכת מהסוג הנ"ל המותקנת במעבדה 116 בבניין הנדסת מכונות (55) באוניברסיטת בן-גוריון בנגב. המערכת כוללת 12 מצלמות המכסות שטח של כ-3x4 מ"ר בגובה של כ-3 מטר. בנפח זה קיים כיסוי מספיק של המצלמות וניתן לעקוב אחרי הסמנים. המצלמות מחוברות למחשב המריץ תוכנה המקליטה את נתוני המיקום של הסמנים וכן מאפשרת הגדרת גוף קשיח מכמה סמנים שונים. הגדרת הגוף הקשיח מאפשרת למקם מספר סמנים על גוף הרובוט ולקבל מיקום ואוריינטציית הרובוט ללא צורך בעיבוד נוסף של מיקומי כל החיישנים. בנוסף להקלטה, המערכת משדרת בזמן אמת את המידע באמצעות UDP Server על רשת האינטרנט. המחשב המריץ את האלגוריתם ומשמש לפיקוד על

הרובוט מריץ UDP Client הממומש בשפת C++ אשר קורא את המידע בזמן אמת ומעביר לתוכנת ה-MATLAB. הקריאות מאפשרות קבלת מיקום ואוריינטציה מצויים בזמן אמת ומאפשרים חישוב ושליחת הפקודות המתאימות לרובוט. בנוסף לקריאות אלו, תוכנת ה-MATLAB קוראת מהרובוט את זוויות כל אחד מהמפרקים. שילוב של כל המידע הנ"ל מאפשר קבלת התנוחה הנוכחית של הרובוט במרחב. השימוש במידע זה נעשה כדי לסגור את חוג בקרת המיקום ולמנוע סחיפות עקב שגיאות מצטברות העולות להיגרם מאי דיוקים ו/או החלקות. לפני כל ביצוע צעד מסט הצעדים שחישוב האלגוריתם, מתקבל המיקום המצוי ומחושבת התנועה שיש לבצע כדי להגיע אל התנוחה הרצויה. איור 43 מציג את הארכיטקטורה המשמשת למעבר המידע.



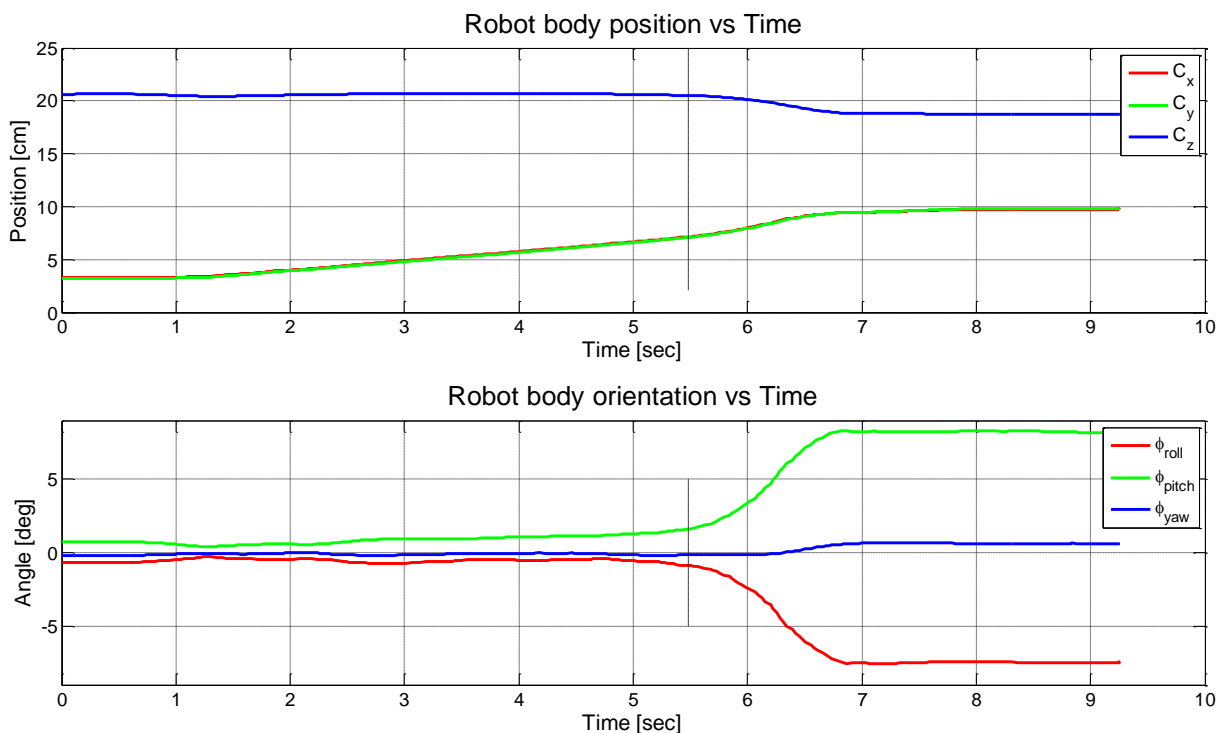
איור 43 - הארכיטקטורה המשמשת למעבר המידע ברובוט

7.2 תוצאות ניסויים

בפרק זה מוצגות תוצאות ניסויים שמטרתם הוכחת היתכנות האלגוריתם. הניסויים המוצגים בוצעו כאשר הקרקע בנויה מלוחות עץ מצופים פורמייקה. מביצוע ניסויים לחישוב מקדם החיכוך נמצא שערכו $\mu = 0.7$. הניסויים בוצעו באמצעות הגדלת שיפוע הקרקע עליו עומד הרובוט, α , עד לרגע ההחלקה ושימוש בקשר $\mu = \tan(\alpha)$. הערך נלקח בממוצע של 30 ניסויים עוקבים כאשר מחצית מהניסויים בוצעו עם מסה שונה.

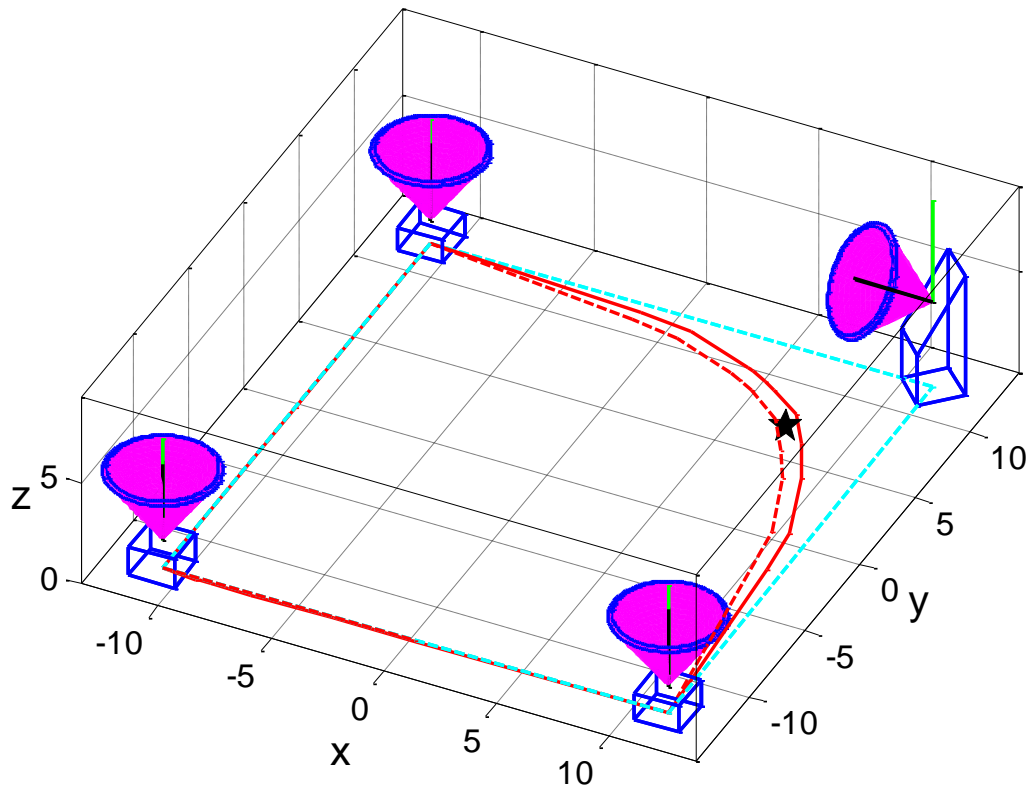
7.2.1 ניסויי וידוא קריטריון יציבות FER

בניסויים אלו מתבצעת בחינה של השימוש בקריטריון ה-FER. במהלך הניסויים הרובוט נדרש להניע את הגוף תוך שמירה על מיקום כפות רגליים קבוע כאשר אחת מרגלי הרובוט דורכת על משטח אשר נמצע בשיפוע 60 מעלות ביחס לאופק. השיפוע מביא להקטנת שטח ה-FER בקרבת נקודת המגע. במהלך הניסויים גוף הרובוט נע תוך הקלטת מיקום הגוף והאוריינטציה בזמן אמת. לאחר כל ניסוי הוצגו ונבחנו הגרפים המתארים את תנועת הגוף ונבחרו הנקודות בהן התבצעה החלקה. דוגמא לגרף עבור אחד הניסויים מוצגת איור 44.



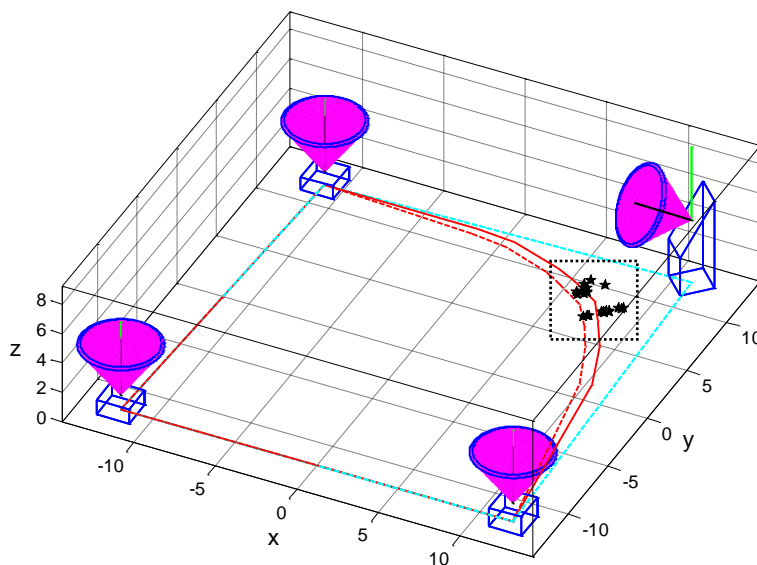
איור 44 – מיקום וזוויות גוף הרובוט כתלות בזמן

כפי שניתן לראות, הגוף נע תוך שמירה על גוף אופקי עד לרגע בו הרובוט מחליק. ניתן לזהות את ההחלקה הן בשינוי החד בזוויות הגלגול (גרף תחתון) והעלרוד והן בירידה בגובה (גרף עליון). באיור 45 ניתן לראות את המיקום על גבי הקרקע המתאים לנקודת ההחלקה שנבחרה.



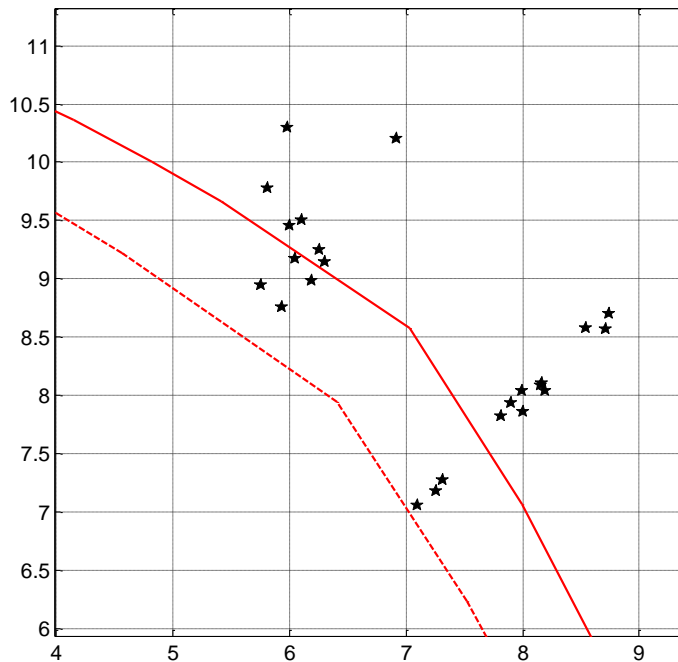
איור 45 - מיקום ההיטל הופקי של מרכז המסה ברגע הנפילה

נקודת ההחלקה (הכוכב השחור) נמצאת על שפת אזור ה- FER המסומנת באדום ולא על שפת ה- SP המסומנת בקווקו תכלת. הקו האדום הרציף (החיצוני) מסמן את האזור ללא מקדם בטחון והקו האדום המקווקו (הפנימי) מסמן את האזור עבור $FSF = 0.9$. בנוסף ניתן לראות באיור המחשה ויזואלית לקונוסי החיכוך בנקודות המגע. בוצעו 24 ניסויים דומים בהם גוף הרובוט נע בכיוונים שונים. התוצאות מוצגות באיור 46.



איור 46 - מיקומי היטלי מרכז המסה ברגע הנפילה ב-24 ניסויים

איור 47 מציג את בהגדלה את האזור הרלוונטי בו התרחשו הנפילות

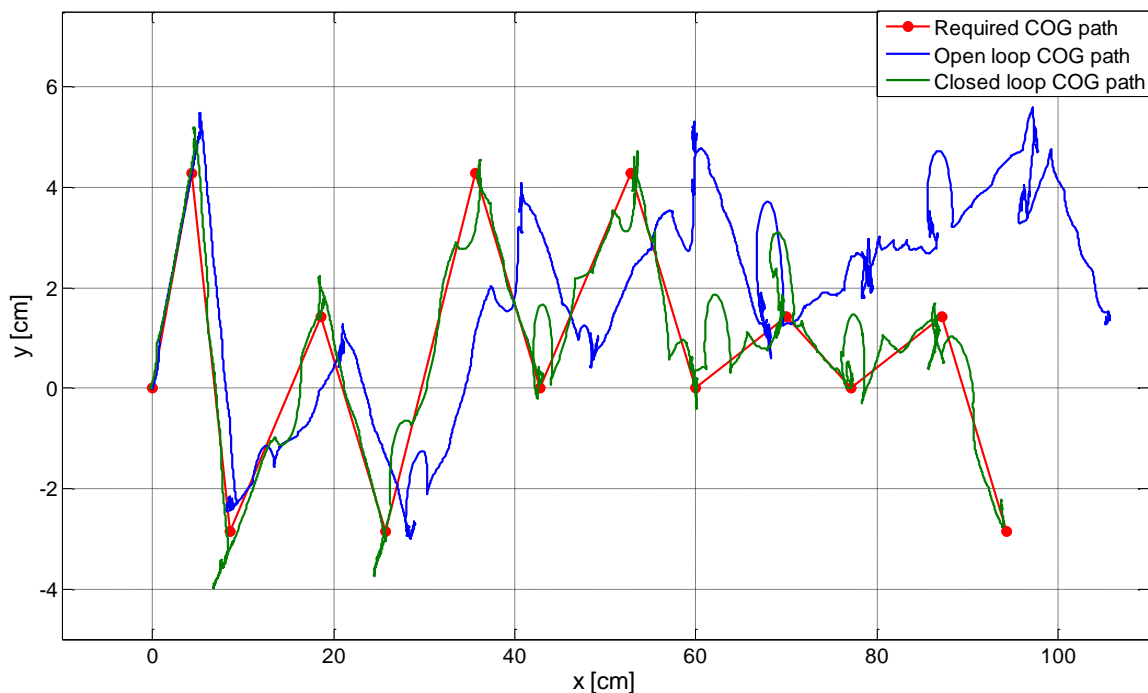


איור 47 - מיקומי היטלי מרכז המסה ברגע הנפילה ב-24 ניסויים - הגדלת אזור הנפילות

כפי שניתן לראות, חלק מהנפילות התרחשו לפני הגעה לשפת האזור שחושב עם המקדם שנמצא בניסוי $\mu = 0.7$ (מסומן בקו אדום רציף). ניתן לראות שמקדם הביטחון שנבחר (קו אדום מקווקו) מביא לקריטריון יציבות אמין יותר כיוון שכל הנפילות מתרחשות מחוץ לגבול שטח ה- FER . המסקנה העיקרית והחשובה מסדרת הניסויים הנ"ל היא שהשימוש בקריטריון היציבות FER הינו הכרחי לצורך תכנון תנועה בקרקע מורכבת.

7.2.2 ניסוי הליכה על קרקע מישורית

ניסוי זה ממחיש הן את היתכנות האלגוריתם והן את אפקטיביות מערכת הניסוי. בניסוי זה הרובוט נדרש ללכת בקו ישר ולעבור מרחק של 1 מטר תוך שמירה על גוף אופקי. ניסוי זה מחולק לשני תתי ניסויים. בראשון הרובוט נשלט בחוג פתוח, כלומר ללא שימוש במערכת לכידת התנועה. הפקודות שקיבל הרובוט נשלחו ברצף ובהנחה שהרובוט התקדם בדיוק כמצופה. תתי הניסוי השני בוצע בחוג סגור. כלומר, בוצע שימוש בערכת לכידת התנועה וקריאת זוויות המפרקים מהרובוט והפקודות שנשלחו לקחו בחשבון את המיקום של הרובוט בפועל. איור 48 מציג את תוצאות שני הניסויים אל מול המסלול הרצוי שהתקבל מהרצת האלגוריתם.



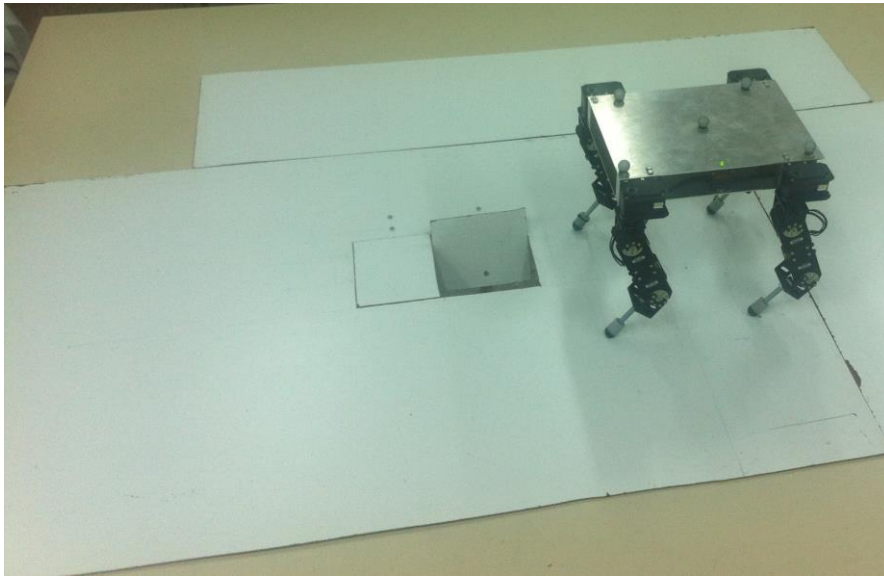
איור 48 - תוצאות ניסויי הליכה בחוג פתוח וחוג סגור

ניתן לראות בבירור שהשימוש בחוג הסגור מביא להגעה ליעד הרצוי בעוד שבחוג הפתוח קיימת שגיאה שהצטברה במהלך ההליכה. בשטחים מורכבים בהם השינויים בקרקע הינם מקומיים וקטנים יש חשיבות רבה לוודאות בנוגע למיקום הרובוט האמיתי ולמיקום כפות רגלי הרובוט. יש לציין שהסיבה לאי הגעת מרכז המסה למרחק 100 ס"מ הינה שנקודת ה-*BP* הוגדרה כמיקום הממוצע של ארבעת כפות הרגליים ולא מיקום מרכז המסה.

הסיבה העיקרית לשגיאות המיקום בעת תנועת הרובוט היא חוסר קואורדינציה בין ארבעת רגלי הרובוט בעת תנועת גוף הרובוט. כאשר כל רגל מבצעת את הפקודה שניתנה לה יתכנו מצבים בהם תנועת רגל אחת מביאה להחלקת רגל אחרת. אומנם ידוע שהמצב ההתחלתי והסופי של תנועת הגוף הוא בעל פתרון קינמטי, אך בעת התנועה בין מצבים אלו רגלי הרובוט אינן מתואמות אחת עם השנייה. החלקות אלו מביאות לסטיות במיקום הרובוט אשר מצטברות תוך כדי ההליכה. פתרון אפשרי הוא קיטוע תנועת הגוף למקטעים קצרים. בניסוי הנ"ל תנועת הגוף חולקה ל-3 מקטעים שווים באורכם. ניתן לראות את השפעת הקיטועים על רציפות התנועה. הרציפות נפגעת מכיוון שהרובוט מגיע לעצירה מלאה בסוף כל מקטע. על מנת להניע את 12 המנועים ממצב אחד למצב אחר קיימת פונקציה הצרובה על הבקר שמקבלת את 12 הזוויות הרצויות ומשך זמן רצוי בו תבצע התנועה. הפונקציה מחשבת את המהירויות הנדרשות לכל מנוע ובאמצעות סגירת חוג עם הזוויות הקיימות מבקרת על מהירויות המנועים עד למצב בו המנועים מגיעים לזוויות הרצויות ונעצרים. עצירה זו משפיעה על רציפות התנועה של הרובוט, כלומר, במקום לנוע באופן רציף בין תנוחה לתנוחה של גוף הרובוט התנועה מתבצעת עם עצירות ביניים. עצירות אלו גורמות לתנודות אשר עלולות לגרום להחלקות של הרובוט על הקרקע. תופעה זו ניתן לפתור באמצעות מימוש חוג בקרת מהירות תוך עקיבה אחר פרופיל מיקום רצוי.

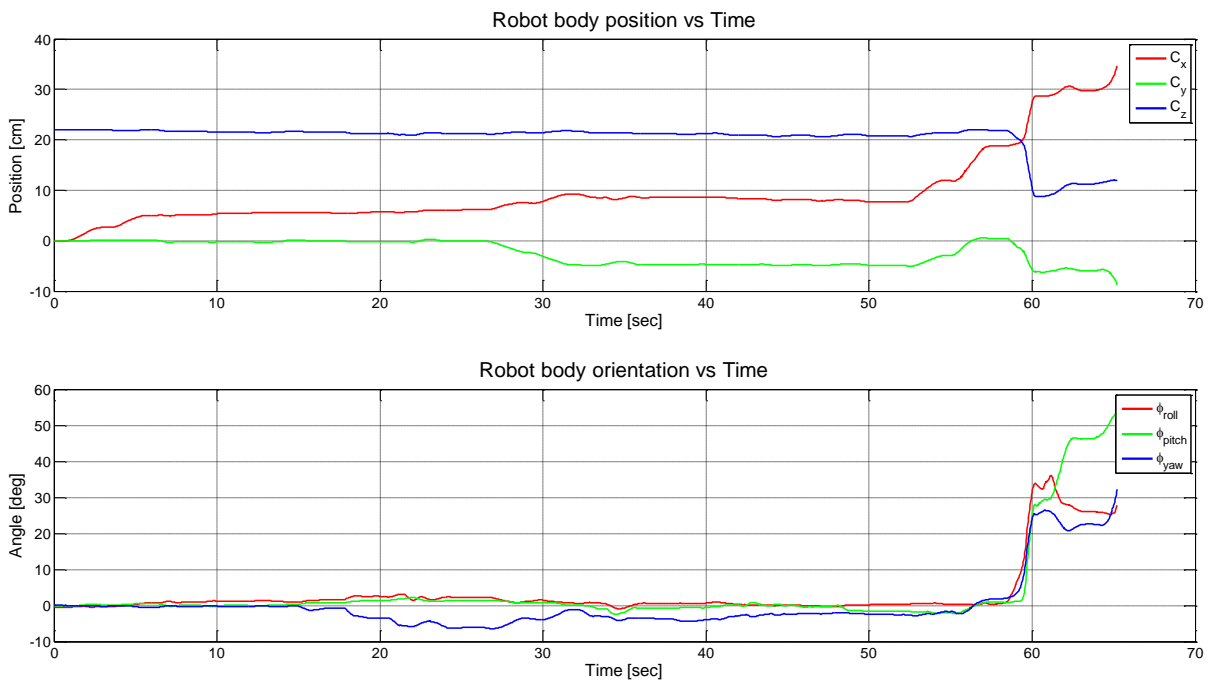
7.2.3 ניסוי הליכה על קרקע מורכבת

תפקיד ניסוי זה הוא להוכיח ולהמחיש את היתרון בשימוש בקריטריון ה-*FER* לתכנון הליכה. הניסוי כולל שני תתי-ניסויים כאשר בראשון הרובוט נדרש לבצע מסלול אשר תוכנן תוך שימוש בקריטריון יציבות *SSM* בלבד. בניסוי השני מתבצע שימוש בפלט של האלגוריתם בשימוש בקריטריון היציבות *FER*. פלטי האלגוריתם בהם בוצע שימוש הם תוצאות סימולציה III (פרק 6.2.3). לצורך הניסוי נבנתה קרקע לפי מודל הקרקע שבסימולציה.



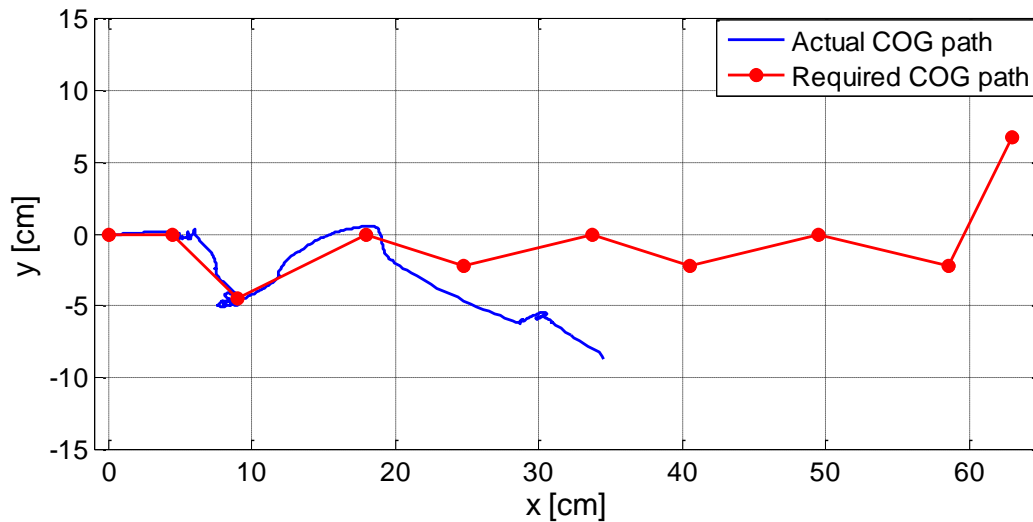
איור 49 - ניסוי הליכה על קרקע מורכבת - קרקע הניסוי

תוצאות הניסוי הראשון מוצגות באיור 50 ובאיור 51.



איור 50 - תוצאות ניסוי הליכה על קרקע מורכבת עם קריטריון *SSM* בלבד – מיקום ואוריינטציית גוף הרובוט כתלות בזמן

ניתן לזהות את הנפילה בבירור לאחר כ-60 שניות. הנפילה מאופיינת בצניחה משמעותית בגובה וכן בשינוי דרסטי בזוויות אוריינטציה הגוף.



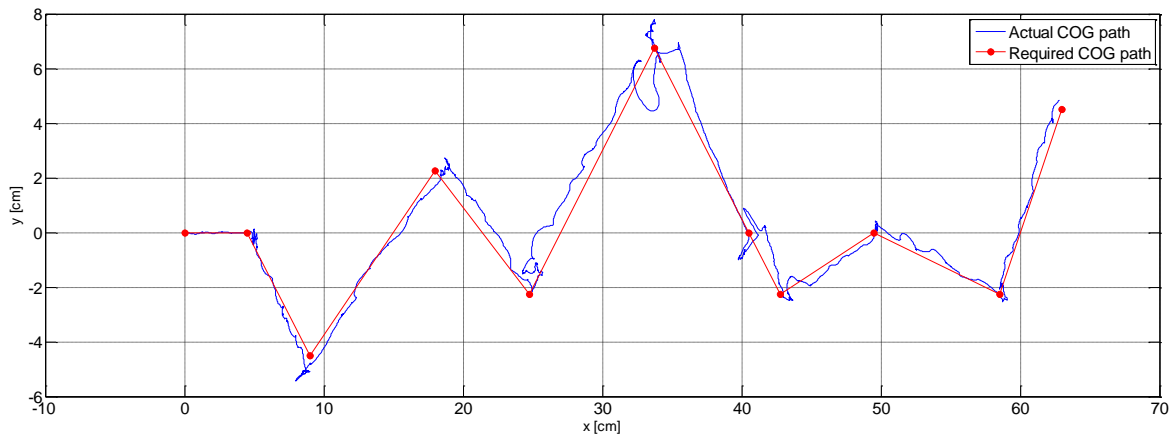
איור 51 - תוצאות ניסוי הליכה על קרקע מורכבת עם קריטריון SSM בלבד – מיקום מצוי אל מול רצוי

התוצאות הנ"ל מציגות את התוצאה שנצפתה בניתוח הסימולציה. ברגע שבו הורמה רגל 3, ראה איור 33(c2), התקבל מצב בו מרכז המסה נמצא מחוץ לאזור ה-*FER* והרובוט החליק ונפל.

תוצאות הניסוי השני בו תנועת הרובוט תוכננה תוך שימוש בקריטריון ה-*FER* מוצגות באיור 52 ובאיור 53.



איור 52 - תוצאות ניסוי הליכה על קרקע מורכבת עם קריטריון FER - מיקום ואוריינטציה גוף הרובוט כתלות בזמן



איור 53 - תוצאות ניסוי הליכה על קרקע מורכבת עם קריטריון *FER* - מיקום מצוי אל מול רצוי

ניתן לראות שכעת המעבר תקין והרובוט אינו נופל. הסיבה העיקרית לסטיות מהמסלול הרצוי היא אותה הסיבה המפורטת בניתוח תוצאות ניסוי ההליכה על הקרקע המישורית. העצירות הנגרמות מקיטועים בתנועת הרובוט גורמות לתנודות ולהחלקות מזעריות בנקודות המגע עם הקרקע. החלקות אלו ניתנות למניעה הן באמצעות מימוש חוגי בקרה מתוחכמים יותר הכוללים בקרת מהירות לאורך מסלול תנועה רצוי ו/או בקרת מומנט והן באמצעות האטת מהירות התנועה של הרובוט ובכך למזער את התנודות הנגרמות בעצירות ובתחילת התנועה. כפי שניתן לראות, גם כאשר קיימות סטיות קלות, השימוש במערכת הבקרה בחוג סגור מאפשר חזרה למסלול הרצוי.

8 סיכום ומסקנות

מטרת מחקר זה הייתה לפתח אלגוריתם גנרי לתכנון הליכה של רובוטים ארבע רגליים. האלגוריתם שפותח מאפשר את תכנון הליכת הרובוט באופן יציב תוך התאמת מאפייני ההליכה למאפייני תנועה רצויים. מתוצאות הסימולציות והניסויים ניתן להתרשם שהשימוש באלגוריתם ככלי לתכנון הליכה הוא פתרון אפשרי ואיכותי. השימוש באלגוריתם מאפשר תכנון של צעדי הרובוט באופן שיבטיח התקדמות אל נקודת היעד תוך שמירה על יציבות הרובוט. המסלול הסופי שמתקבל אינו בהכרח המסלול האופטימלי אך ההתקדמות המקומית, בטווח בו מתבצע חיפוש הגרף עבור סט הצעדים הבאים, היא אופטימלית מבחינת ההתקדמות לעבר המטרה תוך שמירה על יציבות הרובוט.

האלגוריתם מכיל שלב עיבוד נוסף (Post-Processing) אשר מאפשר לבצע ניתוח לתוצאות האלגוריתם ולבצע עקיפת מכשולים במידה ובעיית המינימום המקומי הביאה למסלול בו אין ירידה רציפה בערך הפוטנציאל. שלב זה הינו אחת מהתרומות לחדשנות האלגוריתם.

בעבודה זו שולב קריטריון היציבות *FER* עם אלגוריתם תכנון הליכה. השילוב הוביל לפיתוח של אלגוריתם חדשני בעל יתרון על פני אלגוריתמים דומים אשר אינם מסוגלים לתכנן תנועה על פני קרקע מורכבת. הסימולציות והניסויים שנערכו כחלק ממחקר זה ממחישים את היתרונות בשימוש באלגוריתם. התוצאות והתרומות העיקריות המוצגות בעבודה זו פורסמו והוצגו במספר כנסים [36],[52], [53], [54].

במסגרת המחקר, פותח רובוט הולך ארבע רגלי המשמש ככלי מחקר לחקר רובוטים הולכי ארבע. הרובוט שפותח ונבנה שימש לצורך ביצוע ניסויי היתכנות ובדיקות לאלגוריתם. הרובוט הינו מודולארי ועתיד לשמש ככלי למחקרים עתידיים בתחום.

תרומה ייחודית נוספת היא איחוד התאוריה של תכנון תנועה בין הליכה לטיפוס. האינטגרציה של קריטריון היציבות עם האלגוריתם לתכנון ההליכה קירבה את האלגוריתם לבעל יכולת לתכנן תנועה גם במשימות טיפוס. לצורך השלמת יכולת זו נדרש להוסיף ניתוח לערכי הכוחות המתקבלים מפתרון משוואות קריטריון היציבות.

פיתוח העתידי של האלגוריתם יכלול את שיפור זמן הריצה של האלגוריתם. כדי לבצע תכנון הליכה בזמן אמת יש לשפר את קצב הריצה של האלגוריתם על מנת לייצר הליכה רציפה ללא עצירות. פתרון אפשרי הוא שימוש בעיבוד מקבילי לצורך חיפוש בגרף. תכנון הליכה בזמן אמת דורש מיפוי של הקרקע בזמן אמת. משימה זו אפשרית באמצעות חיישנים כגון סורקי לייזר, מצלמות עומק (נפוץ כחיישן המבוסס על מצלמת וידאו רגילה ומצלמת IR) ועוד. העלות הכלכלית של חיישנים אלו וכן של משאבי מחשוב להרצת האלגוריתם בזמן אמת נמצאת במגמת ירידה ולכן הסבירות לשימוש באלגוריתם לתכנון הליכה בזמן אמת היא גבוהה. פיתוח עתידי נוסף של האלגוריתם יכיל יכולת תכנון הליכה דינמית. הליכה זו מאפשרת תנועה במהירויות גבוהות יותר ומתאימה לתנאי קרקע בהם אין צורך בחישובים מורכבים עבור יציבות הרובוט.

רשימת מקורות

- [1] E. Lucas, "Huitieme Recreation – La Machine a Marcher," *Recreat. Math* 4, pp. 198-204, 1894.
- [2] R.B. McGhee and A.A. Frank, "On the stability properties of quadruped creeping gaits," *Mathematical Bioscience*, pp. 331–351, 1968.
- [3] F. Iida, G. Gomez and R. Pfeifer, "Exploiting body dynamics for controlling a running quadruped robot," in *12th International Conference on Advanced Robotics, 2005. ICAR '05*, 2005, pp. 229 -235.
- [4] A. Shapiro, E. Rimon and S. Shoval, "A Foothold Selection Algorithm for Spider Robot Locomotion in Planar Tunnel Environments," *The International Journal of Robotics Research*, vol. 24, pp. 823-844, 2005.
- [5] A. Sintov, T. Avramovich and A. Shapiro, "Design and motion planning of an autonomous climbing robot with claws," *Robotics and Autonomous Systems*, vol. 59, pp. 1008 - 1019, 2011.
- [6] Little Dog. [Online]. http://www.bostondynamics.com/robot_littledog.html
- [7] M. Hutter, C. Gehring, M. Bloesch, M.A. Hoepflinger, C.D. Remy and R. Siegwart , "StarLETH: A compliant quadrupedal robot for fast, efficient, and versatile locomotion," in *15th International Conference on Climbing and Walking Robot - CLAWAR 2012*, 2012.
- [8] O. Baur, M. Latta, A. Lauber, M. Hutter, M. Hoepflinger, C. Pradalier, R. Siegwart, C. Remy, "Walking and crawling with ALoF: a robot for autonomous locomotion on four legs," *Industrial Robot: An International Journal*, vol. 38, no. 3, pp. 264-268, 2011.
- [9] K. Kato and S. Hirose , "Development of the quadruped walking robot, TITAN-IX - mechanical design concept and application for the humanitarian de-mining robot," *Advanced Robotics*, vol. 15, no. 2, pp. 191-204, 2001.
- [10] T. Kang, H. Kim, T. Son and H. Choi, "Design of quadruped walking and climbing robot," in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 2003, pp. 619-624.
- [11] E. Rimon , S. Shoval and A. Shapiro, "Design of a Quadruped Robot for Motion with Quasistatic Force Constraints," *Autonomous Robots*, vol. 10, pp. 279-296,

- 2001.
- [12] T. Bretl, S. Lall, J.C. Latombe, and S. Rock, "Multi-step motion planning for free-climbing robots," in *The International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2004, pp. 1-16.
 - [13] H. Igarashi, T. Machida, F. Harashima and M. Kakikura, "Free gait for quadruped robots with posture control," in *IEEE International Workshop on Advanced Motion Control*, 2006.
 - [14] H. Igarashi and M. Kakikura, "Path and posture planning for walking robots by artificial potential field method," in *IEEE International Conference on Robotics and Automation*, 2004.
 - [15] J.Z. Kolter, M.P. Rodgers and A.Y. Ng, "A control architecture for quadruped locomotion over rough terrain," in *IEEE International Conference on Robotics and Automation*, 2008.
 - [16] J. Estremera and P.G. de Santos, "Generating continuous free crab gaits for quadruped robots on irregular terrain," *IEEE Transactions on Robotics*, vol. 21, pp. 1067-1076, 2005.
 - [17] J.Z. Kolter and A.Y. Ng, "The Stanford LittleDog: A learning and rapid replanning approach to quadruped locomotion," *Int. J. Rob. Res.*, vol. 30, pp. 150-174, 2011.
 - [18] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry and S. Schaal, "Learning, planning, and control for quadruped locomotion over challenging terrain," *The International Journal of Robotics Research*, vol. 30, pp. 236-258, 2011.
 - [19] J. Buchli, and A.J. Ijspeert, "Self-organized adaptive legged locomotion in a compliant quadruped robot," *Autonomous Robots*, vol. 25, pp. 331-347, 2008.
 - [20] B. Shaoping, K.H. Low, G. Seet and T. Zielinska, "A new free gait generation for quadrupeds based on primary/secondary gait," *In Robotics and Automation*, vol. 2, pp. 1371–1376, 1999.
 - [21] J. Estremera and P. Gonzalez de Santos, "Free Gaits for Quadruped Robots over Irregular Terrain," *The International Journal of Robotics Research*, vol. 21, pp. 115-130, 2002.
 - [22] S. Hirose, "A Study of Design and Control of a Quadruped Walking Vehicle," *The International Journal of Robotics Research*, vol. 3, pp. 113-133, 1984.

- [23] P. Maes and R. Brooks, P. Maes and R.A. Brooks, "Learning to Coordinate Behaviors," 1990.
- [24] J. Dean, T. Kindermann, J. Schmitz, M. Schumm and H. Cruse , "Control of Walking in the Stick Insect: From Behavior and Physiology to Modeling," *Auton. Robots*, vol. 7, pp. 271--288, 1999.
- [25] P.Gonzalez De Santos and M.A. Jimenez, "Path tracking with quadruped walking machines using discontinuous gaits," *Computers & Electrical Engineering*, pp. 383-396, 1995.
- [26] C. Zhang and S. Song, "Gaits and geometry of a walking chair for the disabled," *Journal of Terramechanics*, pp. 211–233, 1989.
- [27] C. Zhang and S. Song, "Stability analysis of wave-crab gaits of a quadruped," *Journal of Robotic Systems*, pp. 243–276, 1990.
- [28] L. Huberty, A comparison of different stability margins for quadruped walking machines, 2008, ETH University, Studies on Mechatronics.
- [29] H. Tsukagoshi and K. Yoneda S. Hirose, "Normalized energy stability margin: generalized stability criterion for walking vehicles," in *Climbing and Walking Robots*, Brussels, 1998, pp. 71–76.
- [30] D. Messuri, "Optimization of the locomotion of a legged vehicle with respect to maneuverability," PhD thesis 1985.
- [31] P. Nagy, "An investigation of walker/terrain interaction," PhD Thesis 1991.
- [32] Y. Or and E. Rimon, "Analytic Characterization of a Class of 3-Contact," *The International Journal of Robotics Research*, pp. 3-22, 2010.
- [33] H. Choset, K.M. Lynch, S. Hutchinson, G.A. Kantor, W. Burgard, L.E. Kavraki, and S. Thrun, *Principles of Robot Motion Theory, Algorithms, and Implementations.:* The MIT Press, 2005.
- [34] Pal, P.K. and Jayarajan, K., "Generation of free gait-a graph search approach," *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 299-305, 1991.
- [35] S. Even, *Graph Algorithms.:* Cambridge University Press, 2011.
- [36] Y. Geva and A. Shapiro, "A Novel Design of a Quadruped Robot for Research Purposes," *International Journal of Advanced Robotic Systems*, (to appear) 2013.
- [37] Dynamixel RX-64 Robot Actuator. [Online].

- http://www.robotis.com/xe/dynamixel_en
- [38] Dynamixel Frames. [Online]. http://www.robotis.com/xe/dynamixel_en
 - [39] ArbotiX Robocontroller. [Online]. <http://www.vanadiumlabs.com/arbotix.html>
 - [40] Arduino. [Online]. <http://arduino.cc/en/>
 - [41] ArbotiX RX-Bridge. [Online]. <http://www.vanadiumlabs.com/rx.html>
 - [42] XBee 2.4 GHz RF Module. [Online]. <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-series1-module>
 - [43] BioloidController library. [Online]. <http://code.google.com/p/arbotix/wiki/BioloidController>
 - [44] ROS - Robot Operating System. [Online]. <http://www.ros.org>
 - [45] K.C. Gupta, "Kinematic Analysis of Manipulators Using the Zero Reference Position Description," *The International Journal of Robotics Research*, vol. 5, no. 2, pp. 5-13, 1986.
 - [46] M.W. Spong, *Robot Modeling and Control*.: WILEY Publication, 2005.
 - [47] R.M. Murray, Z. Li and S.S. Sastry, *A Mathematical Introduction to Robotic Manipulation.*, 1994.
 - [48] A.J.C. Bik and H.A.G. Wijshoff, "Implementation of Fourier-Motzkin Elimination," 1994.
 - [49] T. Bretl and S. Lall, "Testing Static Equilibrium for Legged Robots," *Robotics, IEEE Transactions on*, vol. 24, no. 4, pp. 794-807, 2008.
 - [50] K. Hauser, T. Bretl, J. Latombe, K. Harada, and B. Wilcox, "Motion Planning for Legged Robots on Varied Terrain," *The International Journal of Robotics Research*, vol. 27, pp. 1325-1349, 11-12 2008.
 - [51] J.H. Reif, "Depth-first search is inherently sequential," *Information Processing Letters*, vol. 20, no. 5, pp. 229-234, 1985.
 - [52] Y. Geva and A. Shapiro, "A Combined Potential Function and Graph Search Approach for Free Gait Generation of Quadruped Robots," in *IEEE International Conference on Robotics & Automation (ICRA)*, Minnesota, USA, 2012.
 - [53] Y. Geva and A. Shapiro, "A Combined Potential Function and Graph Search Approach for Free Gait Generation of Quadruped Robots," in *The 32nd Israeli*

- Conference on Mechanical Engineering (ICME)*, Tel-Aviv, Israel, 2012.
- [54] Y. Geva and A. Shapiro, "Quasi-Static Locomotion of Four Legged Robot," in *AUVSI Conference on Unmanned Land Vehicles*, Technion, Israel, 2013.
- [55] V. Loc, I.M. Koo, D.T. Tran, S. Park, H. Moon and H.R. Choi, "Improving traversability of quadruped walking robots using body movement in 3D rough terrains," *Robotics and Autonomous Systems*, vol. 59, no. 12, pp. 1036-1048, 2011.

נספח A – פירוט על פונקציות MATLAB עיקריות

טבלה 3 - פירוט על פונקציות MATLAB עיקריות

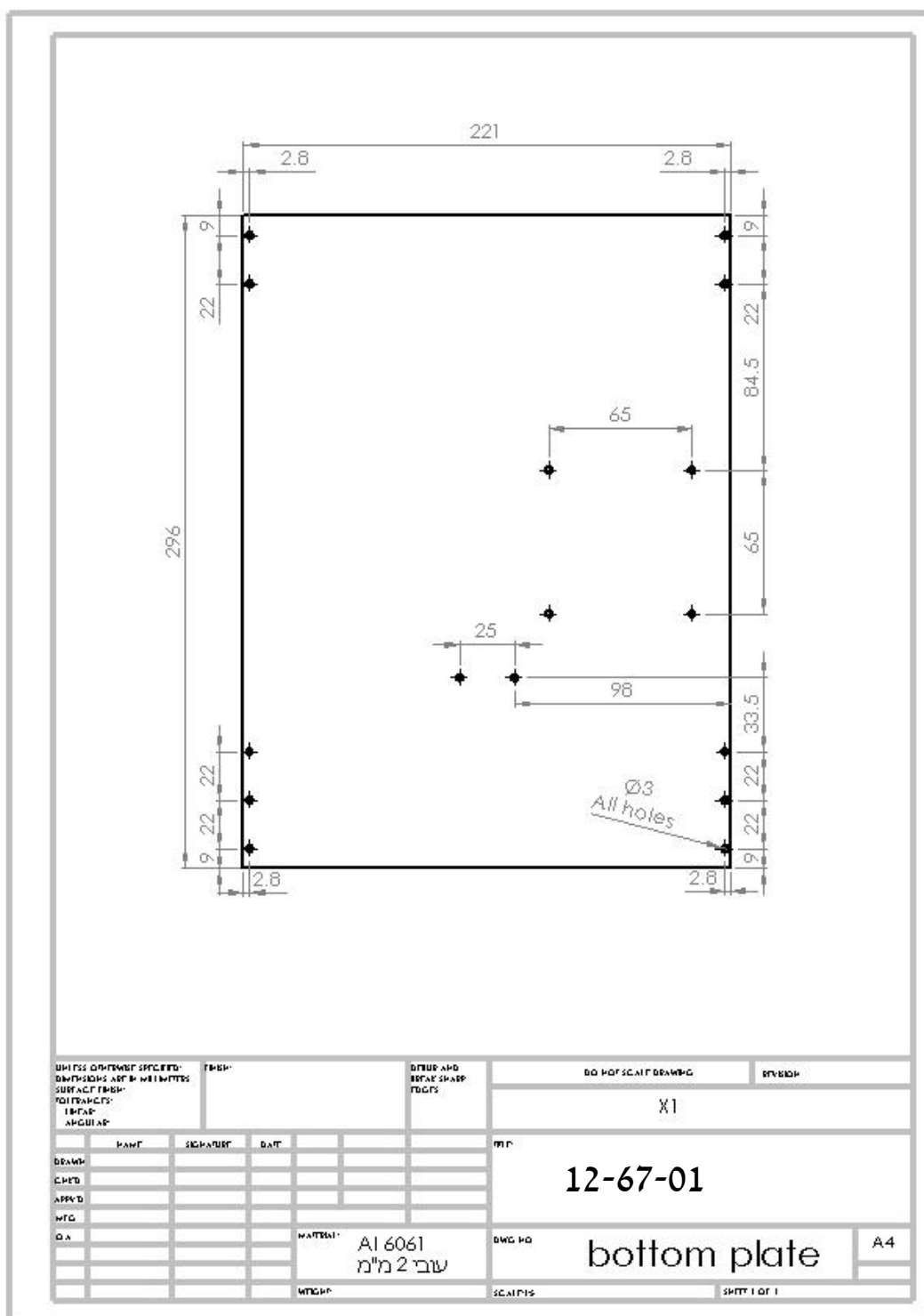
פונקציה	קלט	פלט	תיאור
פונקציות עיקריות למימוש האלגוריתם			
run	<ul style="list-style-type: none"> קובץ הגדרות. 	<ul style="list-style-type: none"> סט הצעדים הסופי לניווט הרובוט. משך זמן כולל וזמן ממוצע לחישוב צעדים. 	<ul style="list-style-type: none"> הפונקציה טוענת את כל ההגדרות הדרושות לריצת האלגוריתם מהקובץ ההגדרות. בפונקציה זו מתבצעת הלולאה הראשית אשר מתבצעת כל עוד הרובוט לא הגיע אל היעד או התקבל מצב ללא פתרון.
step_search	<ul style="list-style-type: none"> מצב הרובוט הנוכחי (וקטור קונפיגורציה). 	<ul style="list-style-type: none"> מצב הרובוט לאחר ביצוע צעד כפול (וקטור קונפיגורציה). פירוט אנדקסי הרגלים שנעו והפוטנציאל המעודכן. 	<ul style="list-style-type: none"> זוהי הפונקציה המבצעת את חיפוש ובחירת הצעד הבא. בפונקציה זו ממומש החיפוש בגרף.
calc_ckw	<ul style="list-style-type: none"> מצב הרובוט הנוכחי (וקטור קונפיגורציה). 	<ul style="list-style-type: none"> מרחב העבודה של הרובוט עבור מצבו הנוכחי (CKW). 	<ul style="list-style-type: none"> הפונקציה מחזירה את כל מיקומי ואוריינטציות הגוף האפשריים עבור מיקם כפות הרגליים הנוכחי.
CKW2CSW	<ul style="list-style-type: none"> מרחב העבודה של הרובוט עבור מצבו הנוכחי (CKW). אינדקס רגל נעה (ML). 	<ul style="list-style-type: none"> מרחב העבודה היציב של הרובוט עבור מצבו הנוכחי (CSW) ותנועת הרגל ML. 	<ul style="list-style-type: none"> הפונקציה מחזירה רק את האיברים ב-CKW אשר מקיימים את תנאי היציבות עבור תנועה של רגל ML.

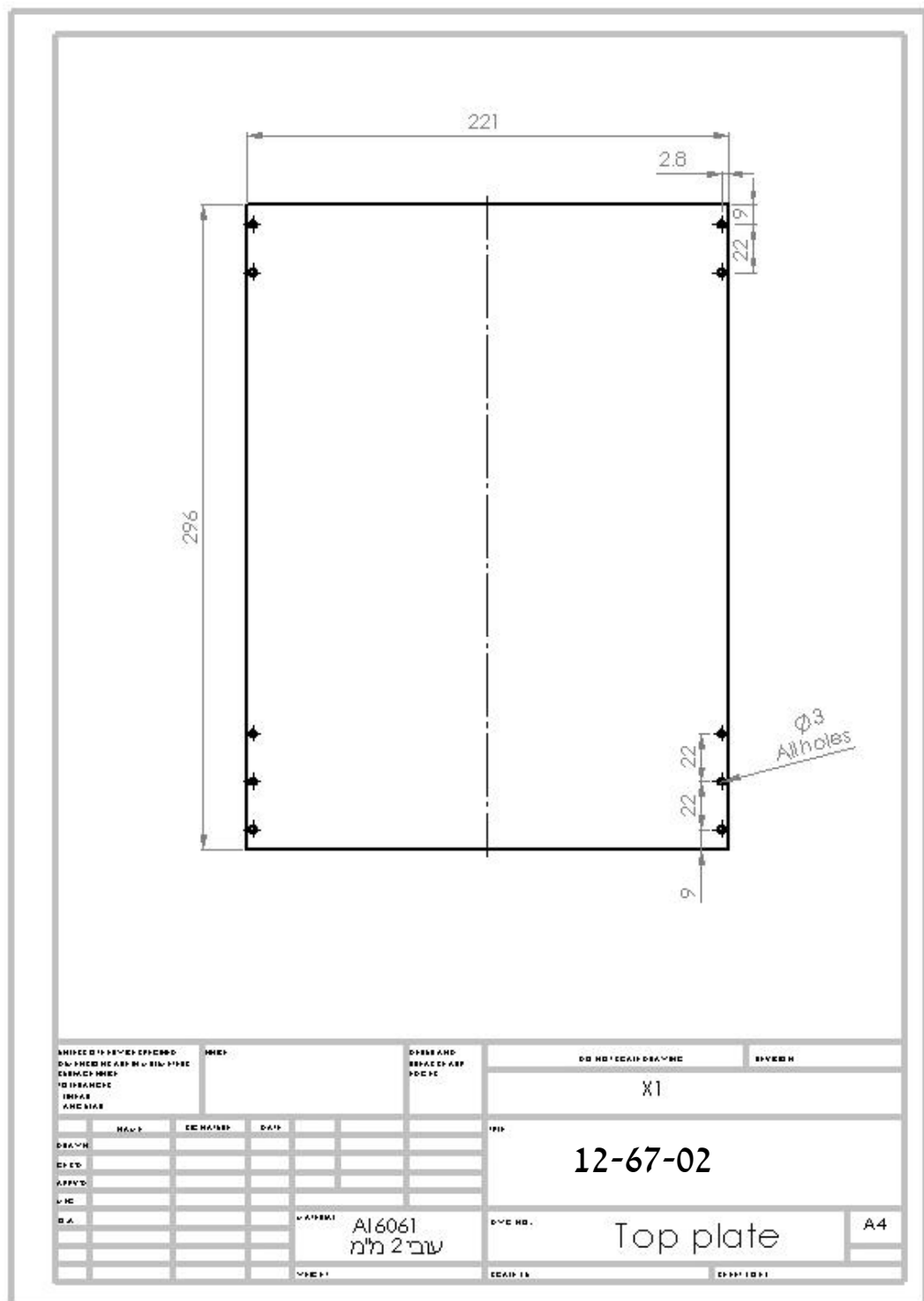
<ul style="list-style-type: none"> • הפונקציה מממשת את קריטריון היציבות <i>FER</i>. 	<ul style="list-style-type: none"> • אישור או שלילה לגבי עמידה בקריטריון שיווי המשקל <i>FER</i>. 	<ul style="list-style-type: none"> • מצב הרובוט (וקטור קונפיגורציה). • רגל נעה (<i>ML</i>). 	isin_FER
<ul style="list-style-type: none"> • הפונקציה מממשת את קריטריון היציבות <i>SSM</i>. 	<ul style="list-style-type: none"> • אישור או שלילה לגבי עמידה בקריטריון שיווי המשקל <i>SSM</i>. 	<ul style="list-style-type: none"> • מצב הרובוט (וקטור קונפיגורציה). • רגל נעה (<i>ML</i>). 	SSM
<ul style="list-style-type: none"> • הפונקציה מחזירה את אוסף הנקודות השייכות לגריד הקרקע הנמצאות בתוך מרחב ה-<i>LWV</i>. 	<ul style="list-style-type: none"> • משטח העבודה של הרגל (<i>LWA</i>). 	<ul style="list-style-type: none"> • מצב הרובוט (וקטור קונפיגורציה). • אינדקס רגל נעה. 	LWA
<ul style="list-style-type: none"> • הפונקציה מממשת את אילוצים A ו-B. 	<ul style="list-style-type: none"> • משטח העבודה של הרגל (<i>LWA</i>) לאחר הסרת נקודות שאינן עומדות באילוצים. 	<ul style="list-style-type: none"> • מצב הרובוט (וקטור קונפיגורציה). • משטח העבודה של הרגל (<i>LWA</i>). • רגל נעה (<i>ML</i>). • רגל הבאה לתנועה (<i>NML</i>). 	ML_Constraints
<ul style="list-style-type: none"> • הפונקציה מממשת את אילוץ C. 	<ul style="list-style-type: none"> • פלט לוגי (True/False) המעיד האם הצעד הנוכחי עומד באילוץ C או לא. 	<ul style="list-style-type: none"> • מצב הרובוט לאחר הצעד הכפול (וקטור קונפיגורציה). • רגל נעה (<i>ML</i>). • רגל הבאה לתנועה (<i>NML</i>). 	Constraint_C
<ul style="list-style-type: none"> • הפונקציה מממשת את אילוץ D באמצעות בדיקה האם הצעד הנוכחי הוא צעד שבוצע בעבר. 	<ul style="list-style-type: none"> • פלט לוגי (True/False) המעיד האם הצעד הנוכחי עומד באילוץ D או לא. 	<ul style="list-style-type: none"> • מצב הרובוט לאחר הצעד הכפול (וקטור קונפיגורציה). • מערך הכולל את כל פרטי הצעדים הקודמים. 	isin_list
<ul style="list-style-type: none"> • הפונקציה מחשבת את ערך פונקציית הפוטנציאל תוך 	<ul style="list-style-type: none"> • ערך פונקציית הפוטנציאל עבור הצעד הנוכחי. 	<ul style="list-style-type: none"> • מצב הרובוט לאחר הצעד הכפול (וקטור 	U_EVAL

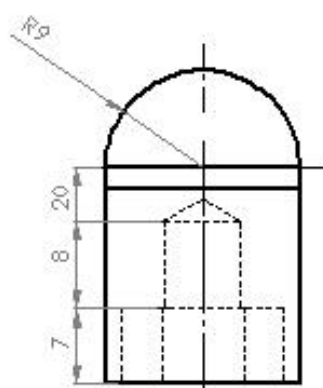
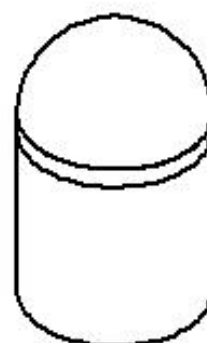
שימוש במשקלים שנקבעו בקובץ ההגדרות.		קונפיגורציה).	
<ul style="list-style-type: none"> הפונקציה מייצרת את רשת גריד הקרקע באמצעות הרכבה של פונקציות שונות הנקבעות מראש. 	<ul style="list-style-type: none"> מערך המכיל את אוסף נקודות גריד הקרקע. 	<ul style="list-style-type: none"> רזולוציית גריד רצויה. אינדקס המשוך לפונקציות הרצויות לשימוש בבניית גריד הקרקע. 	Generate_GROUND
פונקציות עיקריות לביצוע ניסויים			
<ul style="list-style-type: none"> הפונקציה אחראית על פתיחת ערוץ התקשורת החוטית או האלחוטית עם בקר הרובוט. 	<ul style="list-style-type: none"> חיווי על הצלחה או כשל בחיבור 	<ul style="list-style-type: none"> שם הפורט לתקשורת עם הבקר 	arbotix_connect
<ul style="list-style-type: none"> הפונקציה אחראית על סגירת ערוץ התקשורת החוטית או האלחוטית עם בקר הרובוט. 	<ul style="list-style-type: none"> חיווי על הצלחה או כשל בניתוק 	<ul style="list-style-type: none"> שם הפורט לתקשורת עם הבקר 	arbotix_disconnect
<ul style="list-style-type: none"> קריאה לפונקציה זו מתבצעת בעת הגעה של מידע הנשלח מבקר הרובוט אל המחשב. 	<ul style="list-style-type: none"> חיווי לגבי כשל בהגעת נתונים מהבקר. חיווי לגבי סיום תנועת מנועים נוכחית. 	<ul style="list-style-type: none"> מידע שהגיע מהבקר 	arbotix_read
<ul style="list-style-type: none"> זוהי הפונקציה הפותרת את משוואות הקינמטיקה ההפוכה של הרגל. 	<ul style="list-style-type: none"> זוויות שלושת המנועים של הרגל הנעה המתאימות למצב הרובוט הרצוי. 	<ul style="list-style-type: none"> מצב הרובוט (וקטור קונפיגורציה). רגל נעה (ML). 	Analitic_LEG_IK
<ul style="list-style-type: none"> הפונקציה ממירה את רצף מצבי הרובוט לרץ של זוויות מנועים. 	<ul style="list-style-type: none"> סט זוויות מנועים התואם לסט הצעדים הסופי הרצוי. 	<ul style="list-style-type: none"> סט הצעדים הסופי לניווט הרובוט. 	Walk
<ul style="list-style-type: none"> פונקציה זו מתכננת את מסלול התנועה של מנועי הרובוט המביא לתנועה של 	<ul style="list-style-type: none"> סט זוויות מנועים אשר מביא לתנועת גוף הרובוט בין שני מצבי רובוט עוקבים. 	<ul style="list-style-type: none"> שני מצבי רובוט עוקבים. 	Move_COG

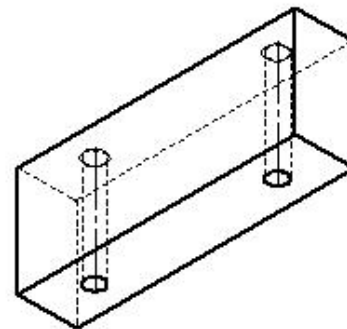
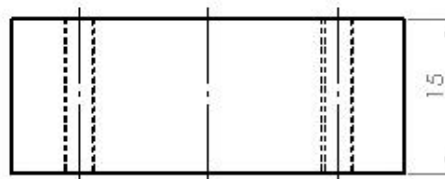
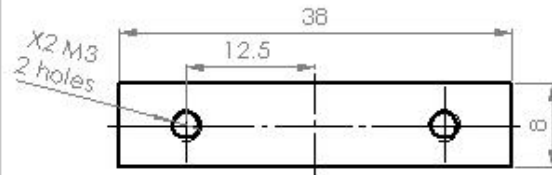
גוף הרובוט בין שני מצבים עוקבים.			
<ul style="list-style-type: none"> פונקציה זו מתכננת את מסלול התנועה של רגלי הרובוט בעת ביצוע הצעדים (הרמת והורדת רגליים). 	<ul style="list-style-type: none"> סט זוויות מנועים אשר מביא לתנועת שני רגלי הרובוט (רגל ראשונה ושנייה בצעד כפול) בין שני מצבי רובוט עוקבים. 	<ul style="list-style-type: none"> שני מצבי רובוט עוקבים. 	Move_LEGS

נספח B – שרטוטי חלקים שתוכננו לשם בניית הרובוט





[illegible]



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: HORIZONTAL: ANGULAR:				FINISH:		OTHER AND BOTH SHARP EDGES		DO NOT SCALE DRAWING		REVISION	
DRAWN				SIGNATURE		DATE		DATE			
CHECKED								12-67-04			
APPROVED								DWG NO			
MTC								battery holder 2			
Q.A.								A4			
								MATERIAL			
								AI 6061			
								SCALE			
								SHEET 1 OF 1			

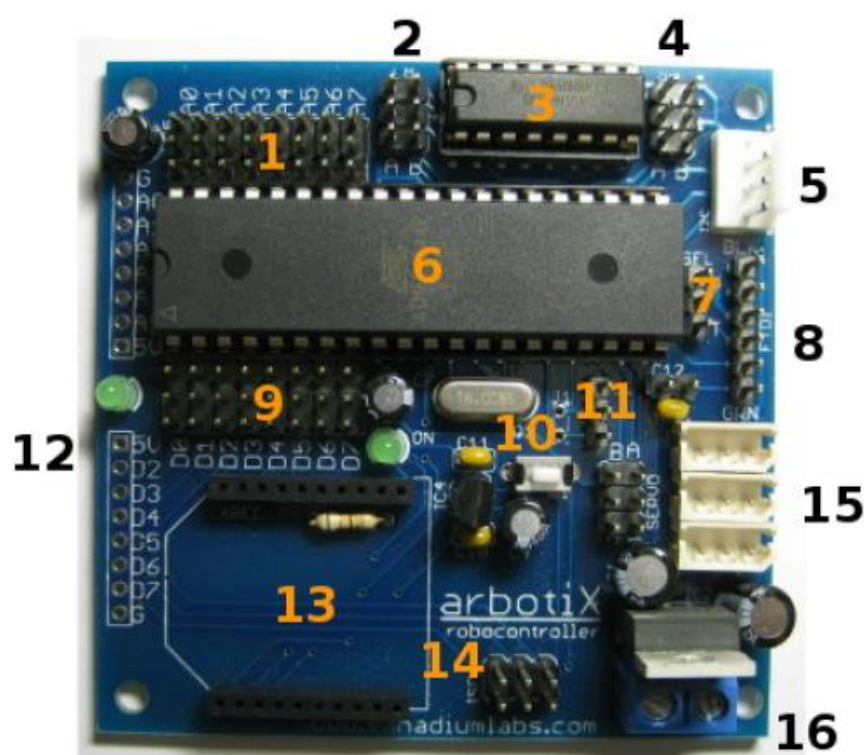


1-3. Specifications of RX-64

	RX-64	
Weight (g)	125	
Dimension (mm)	40.2 x 61.1 x 41.0	
Gear Reduction Ratio	1/200	
Applied Voltage (V)	at 15V	at 18V
Final Reduction Stopping Torque (kgf.cm)	64.4	77.2
Speed (Sec/60 degrees)	0.188	0.157

Resolution	0.29°
Running Degree	300°, Endless Turn
Voltage	12V~21V (Recommended voltage: 18V)
Max Current	1200mA
Running Temperature	-5°C ~ +85°C
Command Signal	Digital Packet
Protocol	RS485 Asynchronous Serial Communication (8bit, 1stop, No Parity)
Link (Physical)	RS485 Multi Drop Bus
ID	254 ID (0~253)
Communication Speed	7343bps ~ 1 Mbps
Sensing & Measuring	Position, Temperature, Load, Input Voltage, etc.
Material Quality	Full Metal Gear, Engineering Plastic Body
Motor	Maxon RE-MAX
Standby Current	50 mA

נספח D – מפרט בקר ARBOTIX



- | | |
|---------------------------------------|---|
| 1 - Analog port headers | • 9 - Digital port headers |
| 2 - Left motor/encoder headers | • 10 - Reset Switch |
| 3 - Dual motor driver, max current 1A | • 11 - Serial1 header (also J1) |
| 4 - Right motor/encoder headers | • 12 - Prototyping headers and user led |
| 5 - I2C header | • 13 - XBEE socket |
| 6 - ATMEGA644P | • 14 - In-system programming (ISP) |
| 7 - Power selection header | • 15 - 3 Bioid headers |
| 8 - FTDI serial0/programming | • 16 - Power terminals |

1. XBee®/XBee-PRO® RF Modules

The XBee and XBee-PRO RF Modules were engineered to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between devices.

The modules operate within the ISM 2.4 GHz frequency band and are pin-for-pin compatible with each other.



Specifications

Table 1-01. Specifications of the XBee®/XBee-PRO® RF Modules

Specification	XBee
Performance	
Indoor/Urban Range	Up to 100 ft (30 m)
Outdoor RF line-of-sight Range	Up to 300 ft (90 m)
Transmit Power Output (software selectable)	1mW (0 dBm)
RF Data Rate	250,000 bps
Serial Interface Data Rate (software selectable)	1200 bps - 250 kbps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)
Power Requirements	
Supply Voltage	2.8 – 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)
Idle / Receive Current (typical)	50mA (@ 3.3 V)
Power-down Current	< 10 μ A
General	
Operating Frequency	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)
Operating Temperature	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector, RP-SMA Connector
Networking & Security	
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer
Number of Channels (software selectable)	16 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses
Agency Approvals	
United States (FCC Part 15.247)	OUR-XBEE
Industry Canada (IC)	4214A XBEE
Europe (CE)	ETSI
Japan	R201WW07215214
Australia	C-Tick